



ITMO UNIVERSITY

Saint Petersburg, Russia

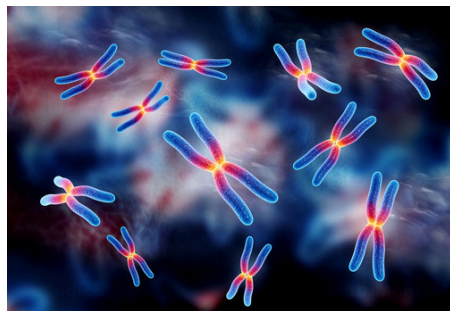
Lecture 1: Introduction to Evolutionary Computing

Michael Melnik

mihail.melnik.ifmo@gmail.com

What is Evolutionary Computing?

- ✓ Evolutionary Computing (EC) is a field of **stochastic optimization** techniques used to search for an optimal solution through a multidimensional solution spaces.
- ✓ Evolutionary Computing is a part of Soft Computing
- ✓ Almost all Evolutionary Algorithms (EAs) are nature inspired



What is the optimization?

- ✓ Optimization has the goal of constructing and analyzing approximate solutions to problems which cannot be solved exactly
- ✓ Optimization algorithms use smart ways of choosing input values of a system and evaluate output values
- ✓ The search takes place within given constraints and optimization criteria

Optimization definition

$\vec{X} = [x_1, x_2, \dots, x_D]^T$ parameter vector, input

$f(\vec{X})$ system values, output

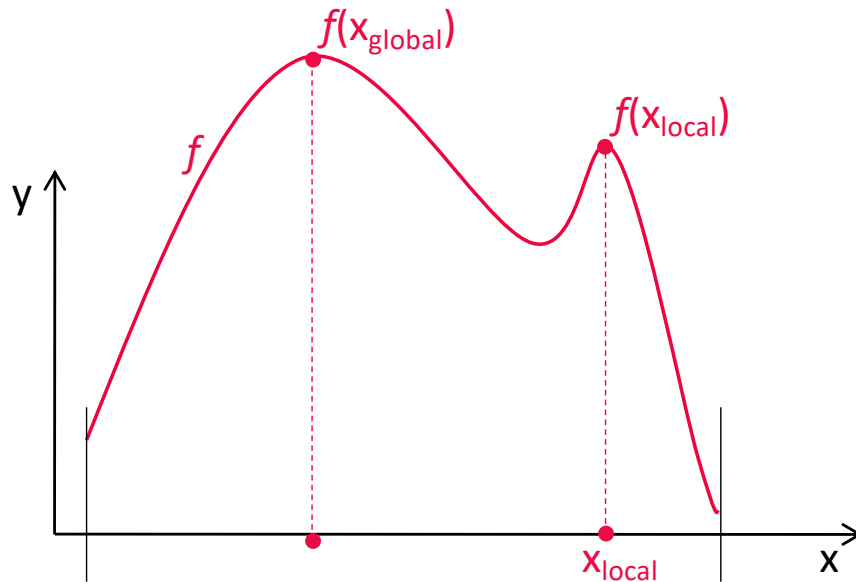
f objective function: $\mathbb{R}^D \xrightarrow{f} \mathbb{R}$



We are looking for $\vec{X}^* \in \mathbb{R}^D$ that minimizes the objective function f .
or $f(\vec{X}^*) < f(\vec{X})$ for all $\vec{X} \neq \vec{X}^*$.

Finding a maximum is the same, because $\max\{f(\vec{X})\} = -\min\{-f(\vec{X})\}$

Global vs local optimization



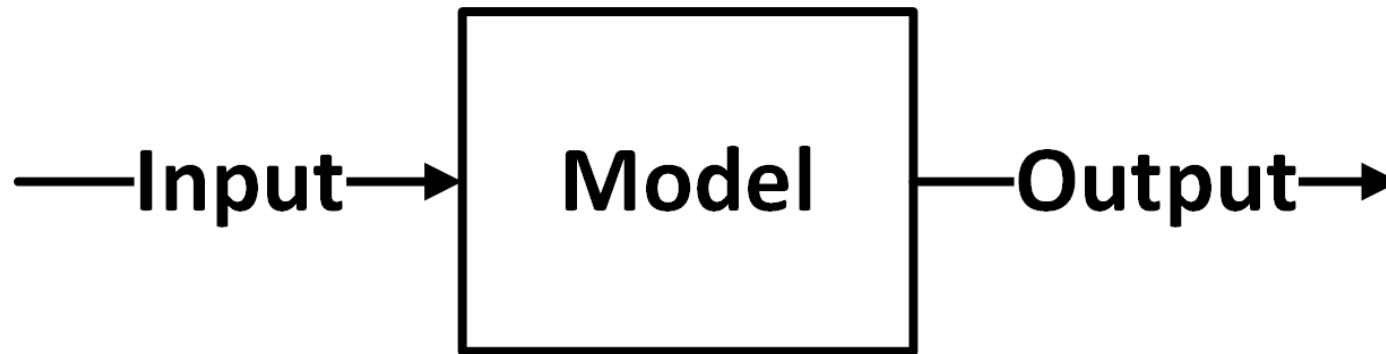
- ✓ Global optimization is **more difficult** than local optimization.
- ✓ Optimization algorithm should have a possibility to **avoid local optima**.
- ✓ But not always.

Examples of applications

Problems can be classified in different ways:

- Black box model
- Search problems
- Optimization vs constraint satisfaction
- NP problems

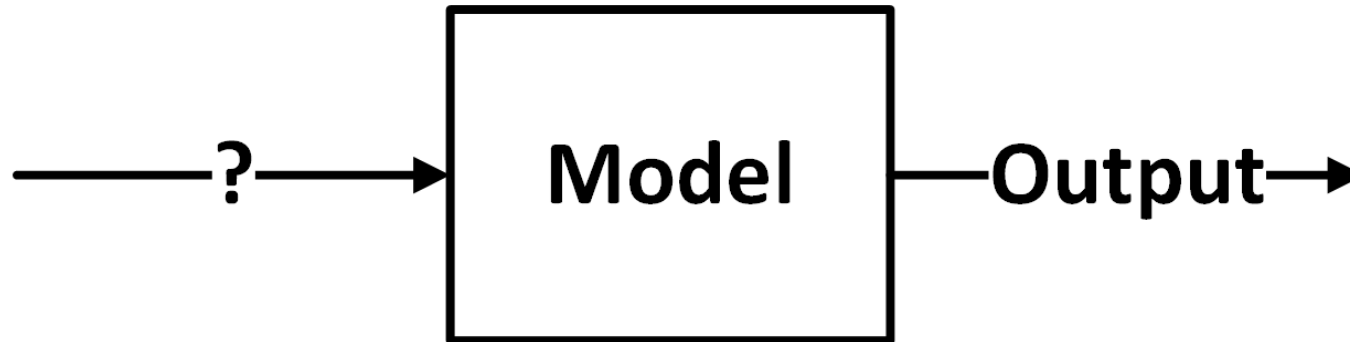
“Black box” optimization



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

“Black box” model: optimization

- ✓ Model and desired output is known, task is to **find inputs**



- ✓ Examples:
 - Time tables for university, schools, hospitals,...
 - Traveling salesman problem
 - Eight-queens problem

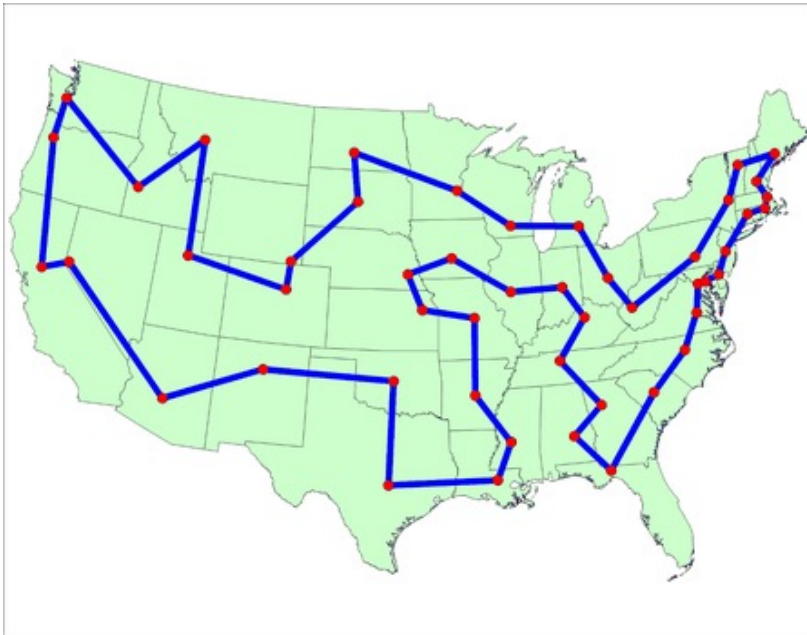
University timetable



- ✓ 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

	Monday, October 10	Tuesday, October 11	Wednesday, October 12	Thursday, October 13	Friday, October 14
9	Fundamentals of Bioinformatics Hoorcollege	Fundamentals of Bioinformatics Hoorcollege		Scientific Visualization and Virtual Reality Hoorcollege Machine Learning 1 Workcollege	Internet programming Hoorcollege
10					
11	Fundamentals of Bioinformatics Computerpracticum	Machine Learning 1 Workcollege (3)	Fundamentals of Bioinformatics Computerpracticum	Introduction to Computational Science Laptopcollege Non-Linear Economics	Applied Mechanism Design and Big Data Hoorcollege
12				Statistical Theory of Complex Molecular Systems Hoorcollege Finite Element Method: Theory and Applications Hoorcollege	Statistical Theory of Complex Molecular Systems Hoorcollege Finite Element Method: Theory and Applications Hoorcollege
13	Programming Large-scale Parallel Systems Hoorcollege	Statistical Theory of Complex Molecular Systems Hoorcollege Machine Learning 1 Laptopcollege (2)	Machine Learning 1 Hoorcollege Finite Element Method: Theory and Applications Laptopcollege	Stochastic Processes for Finance Hoorcollege	Finite Element Method: Theory and Applications Laptopcollege
14		Statistical Theory of Complex Molecular Systems Hoorcollege Machine Learning 1 Laptopcollege (2)	Machine Learning 1 Hoorcollege Finite Element Method: Theory and Applications Laptopcollege	Evolutionary Computing Hoorcollege Programming Large-scale Economics	Introduction to Computational Science Workcollege Non-Linear Economic Dynamics Computerpracticum
15		Statistical Theory of Complex Molecular Systems Hoorcollege Machine Learning 1 Laptopcollege (2)	Machine Learning 1 Hoorcollege Finite Element Method: Theory and Applications Laptopcollege	Internet programming Hoorcollege	Seminars Computational Science Presentatie
16		Statistical Theory of Complex Molecular Systems Hoorcollege Machine Learning 1 Laptopcollege (2)	Machine Learning 1 Hoorcollege Finite Element Method: Theory and Applications Laptopcollege		
17	Machine Learning 1 Workcollege Applied Mechanism Design and Big Data Hoorcollege	Machine Learning 1 Laptopcollege (2)	Machine Learning 1 Laptopcollege (2)		

Traveling salesman problem (TSP)



- ✓ Given N cities with distances between them
- ✓ Find the **shortest path** across all cities



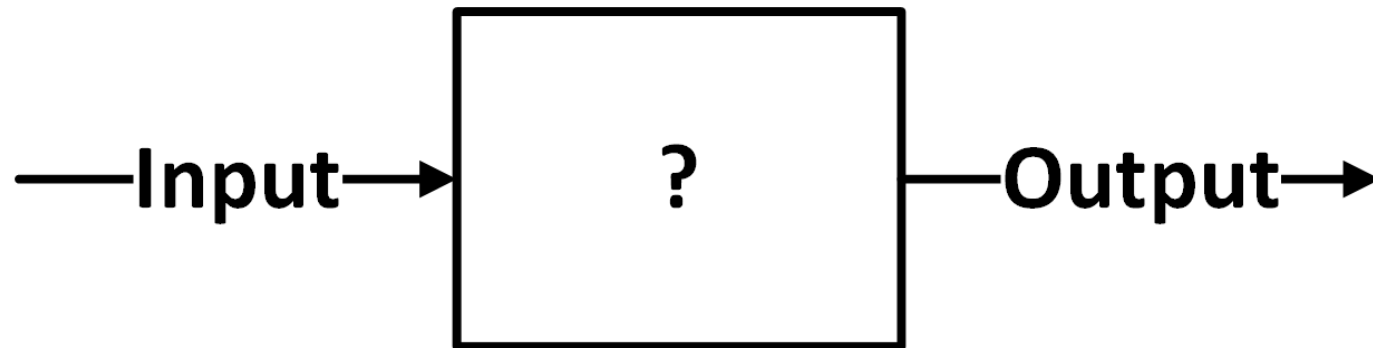
Queens problem

- ✓ 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 ($n > 8$)



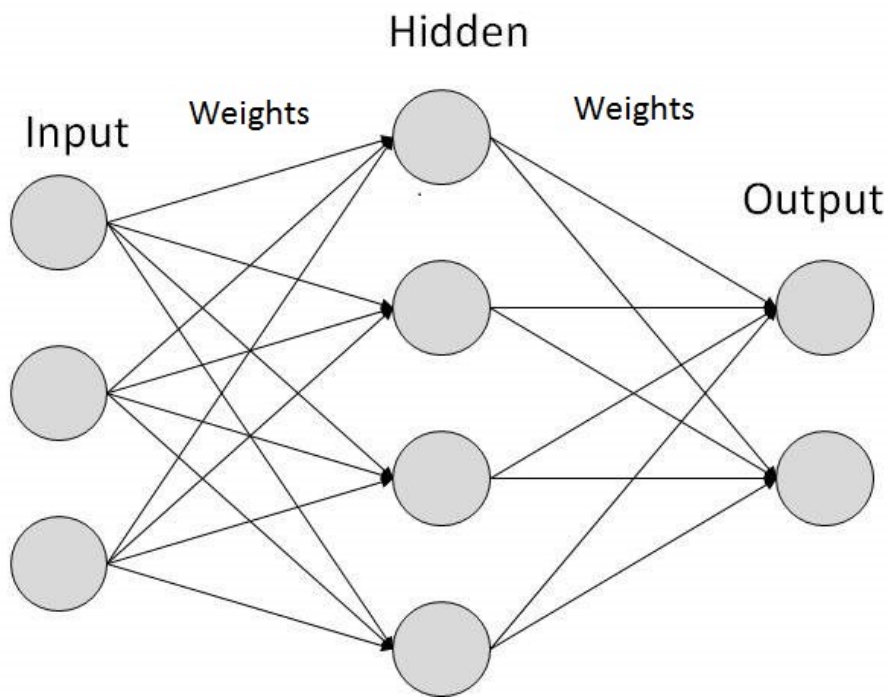
“Black box” model: modelling

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



- ✓ Note: modelling problems can be transformed into optimisation problems
 - Evolutionary machine learning

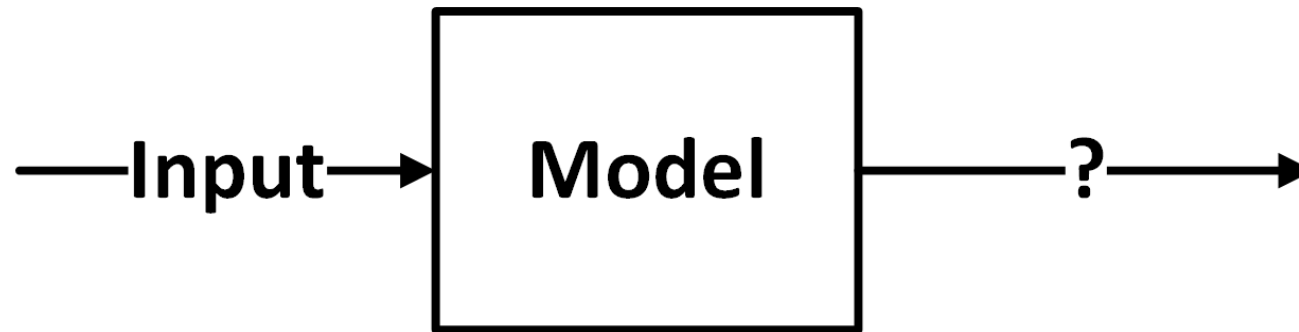
Neural networks training



- ✓ Learning of net's parameters
- ✓ Suitable for reinforcement learning

“Black box” model: 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

Search problems

- ✓ Simulation is **different** from optimization/modelling
- ✓ Optimization/modelling problems **search** through huge space of possibilities
- ✓ Search space: **collection** of all objects of interest including the desired solution

Benefit of classifying these problems: distinction between

- search problems, which define search spaces, and
- problem-solvers, which tell how to move through search spaces.

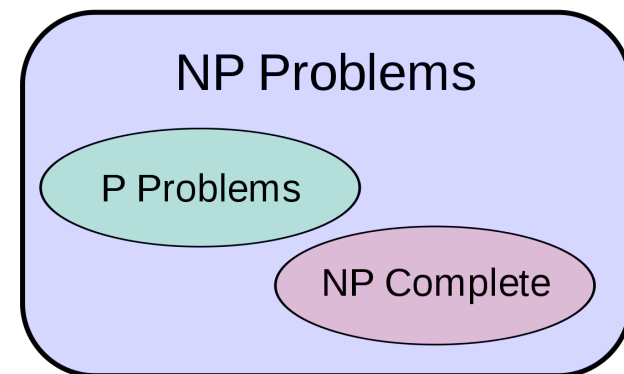
Optimization vs. constraint satisfaction

- ✓ Objective function: a way of assigning a value to a possible solution that reflects its quality on scale
 - Number of un-checked queens (maximize)
- ✓ Constraint: binary evaluation telling whether a given requirement holds or not
 - Find a configuration of eight queens on a chessboard such that no two queens check each other

	Objective function	
Constraints	Yes	No
Yes	Constrained optimization problem	Constrained satisfaction problem
No	Optimization problem	No problem

NP problems

- ✓ This classification scheme needs the properties of the problem solver
- ✓ Benefit of this scheme: possible to tell how difficult the problem is
- ✓ Explain the basics of this classifier for combinatorial optimization problems (booleans or integers search space)



NP problems: key notions

- ✓ **Problem size**: dimensionality of the problem at hand and number of different values for the problem variables
- ✓ **Running-time**: number of operations the algorithm takes to terminate
 - Worst-case as a function of problem size
 - Polynomial, super-polynomial, exponential
- ✓ **Problem reduction**: transforming current problem into another via mapping

NP problems: class

- ✓ The difficultness of a problem can now be classified:
 - **Class P**: algorithm can solve the problem in polynomial time (worst-case running-time for problem size n is less than $F(n)$ for some polynomial formula F)
 - **Class NP**: problem can be solved and any solution can be verified within polynomial time by some other algorithm (P subset of NP)
 - **Class NP-complete**: problem belongs to class NP can be reduced to this problem by algorithm running in polynomial time
 - **Class NP-hard**: problem is at least as hard as any other problem in NP-complete but solution cannot necessarily be verified within polynomial time

Motivation for EC

- ✓ Developing, analyzing, applying problem solving methods is central theme in mathematics and computer
- ✓ Complexity of problems to be solved increases
- ✓ Global optimization problems
- ✓ Multi-objective optimization problems
- ✓ Combinatorial or fuzzy representation of problem
- ✓ It's funny

- ✓ The best problem solver known in nature:
 - Human brain -> Machine Learning
 - Evolution mechanism -> Evolutionary Computing

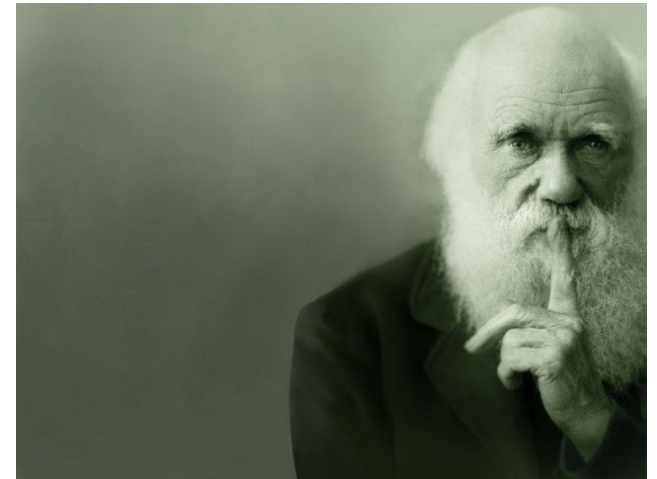
Brief History of EC

- 1948, Turing
proposes “genetical or evolutionary search”
- 1962, Bremermann
optimization through evolution and recombination
- 1964, Rechenberg
introduces evolution strategies
- 1965, L. Fogel, Owens and Walsh
introduce evolutionary programming
- 1975, Holland
introduces genetic algorithms
- 1992, Koza
introduces genetic programming

GENETIC ALGORITHM

✓ Pre-notes:

- EC is part of computer science
- EC is not part of life sciences/biology
- Biology delivered inspiration and terminology
- EC can be applied in biological research



Darwinian Evolution: “Survival of the fittest”

- All environments have finite resources
(i.e., can only support a limited number of individuals)
- Life forms have basic instinct/ lifecycles geared towards reproduction
- Therefore some kind of selection is inevitable
- Those individuals that compete for the resources most effectively have increased chance of reproduction

Mutation

- ✓ Darwin: “small, random variations in phenotypic traits occur during reproduction from generation to generation.
- ✓ Occasionally some of the genetic material changes very slightly during this process (replication error)
- ✓ This means that the child might have genetic material information not inherited from either parent
- ✓ This can be
 - catastrophic: offspring is not viable (most likely)
 - neutral: new feature does not influence fitness
 - advantageous: strong new feature occurs

Darwinian Evolution

- ✓ Population consists of diverse set of individuals
- ✓ Combinations of traits that are better adapted tend to increase representation in population

Individuals are “units of selection”

- ✓ Variations, selection lead to evolution

Population is the “unit of evolution”

- ✓ There is no “guiding force”



The main EC Metaphor

✓ Evolution

- Environment
- Individual
- Fitness

✓ Problem Solving

- Problem
- Candidate solution
- Quality

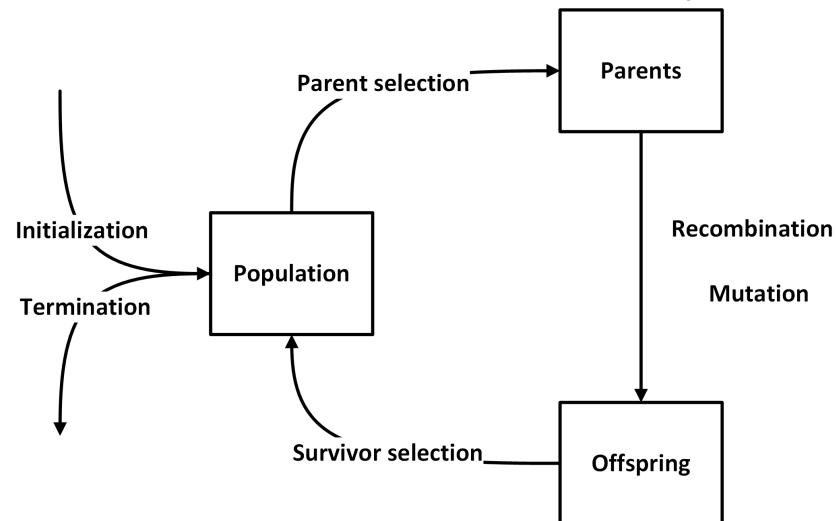
Fitness – chances for survival and reproduction

Quality – chance for seeding new solutions

What is an Evolutionary Algorithm?

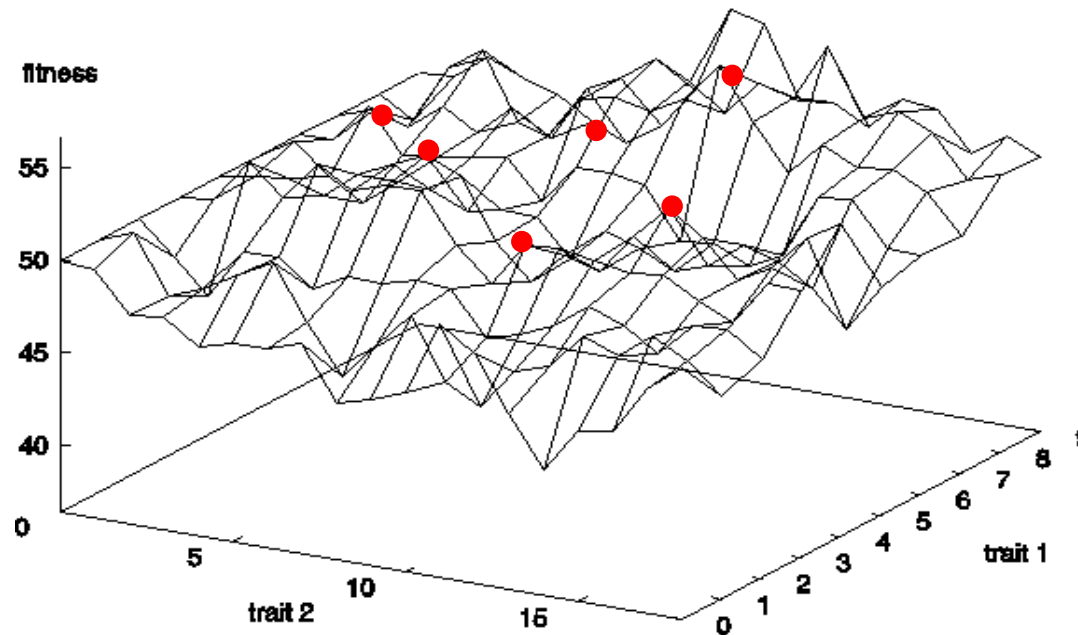
General Description

- ✓ EAs fall into the category of “generate and test” algorithms
- ✓ They are stochastic, population-based algorithms
- ✓ Variation operators (recombination and mutation) create the necessary diversity



Fitness landscape

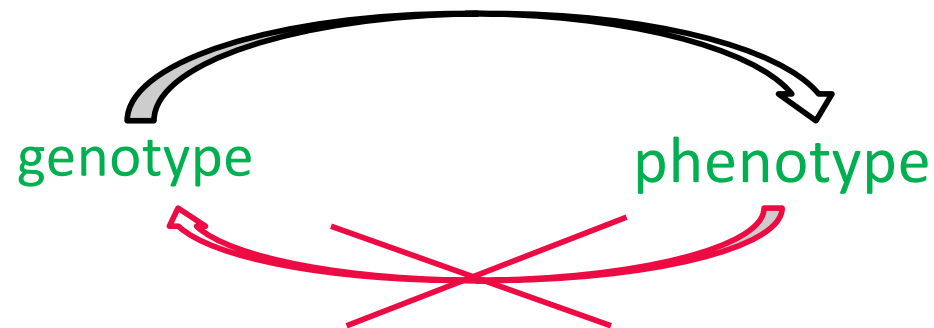
- fitness landscape, or adaptive landscape
- population is seen as a cloud of points in this landscape
- individual = candidate solution is one point



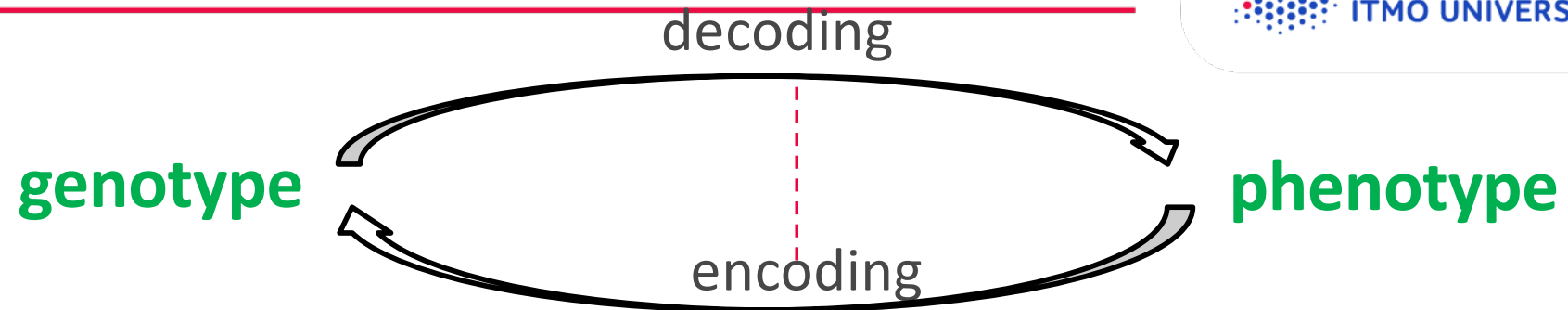
Genetics

- ✓ The information required to build a living organism is coded in the DNA of that organism
- ✓ Genotype (DNA inside) determines phenotype (actual observation)
- ✓ Genes → phenotypic traits is a complex mapping
 - One gene may affect many traits (*pleiotropy*)
 - Many genes may affect one trait (*polygeny*)

In nature



in EC:

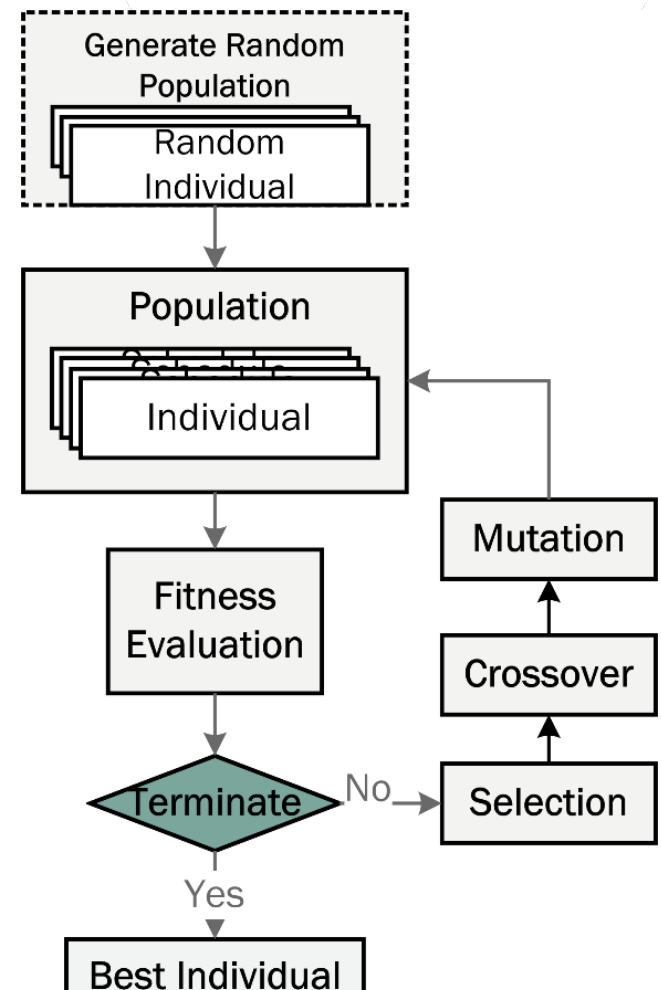
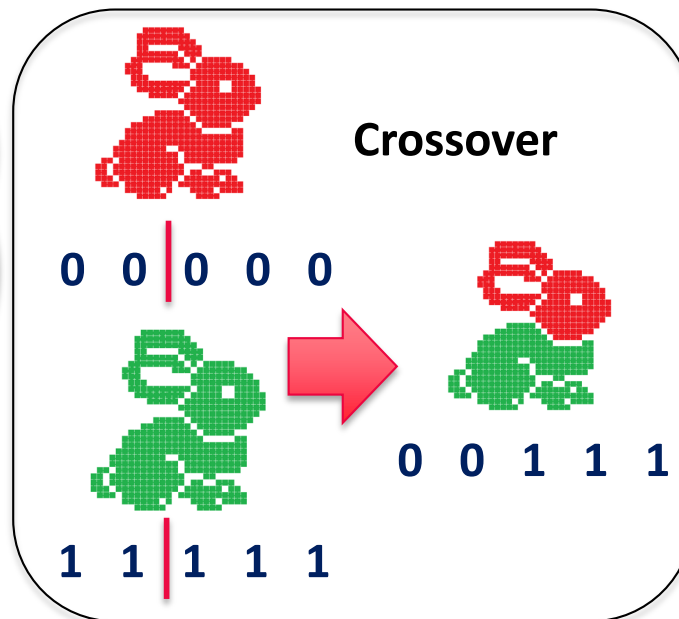
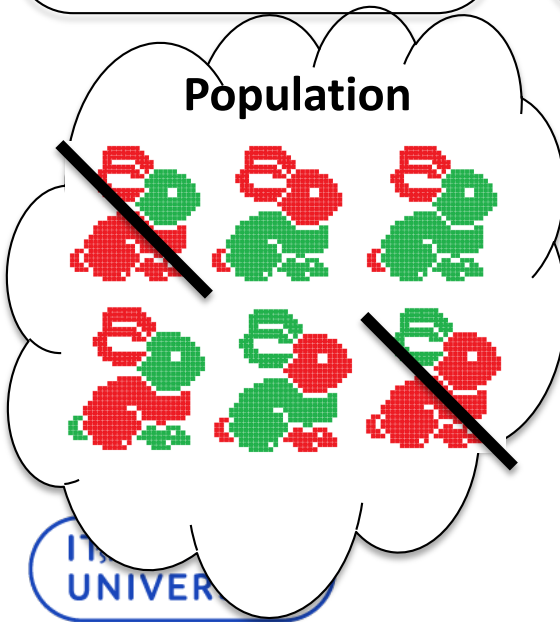
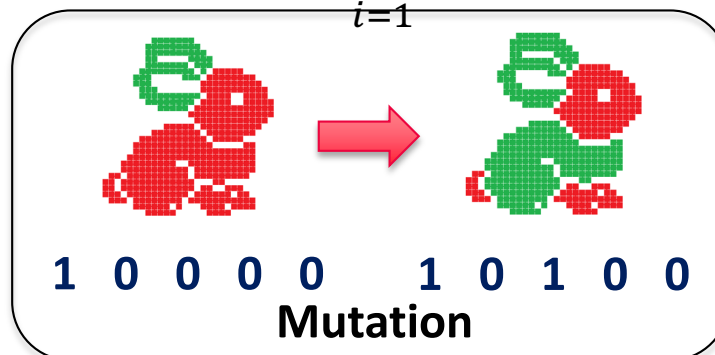
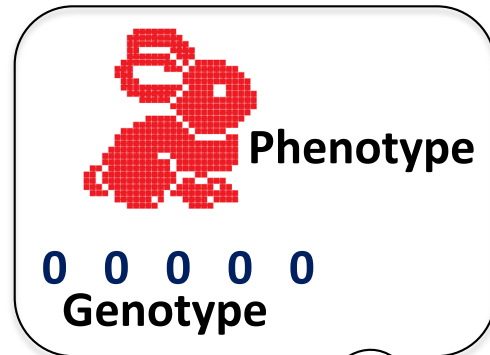


code
inside
evolutionary variations
(mutation, recombination)
chromosome
point in genotype space

features & behaviour
outside
environment
selection
survival
candidate solution
point in phenotype space
fitness → problem

GA example

$$Fitness = \sum_{i=1}^n Genotype_i$$



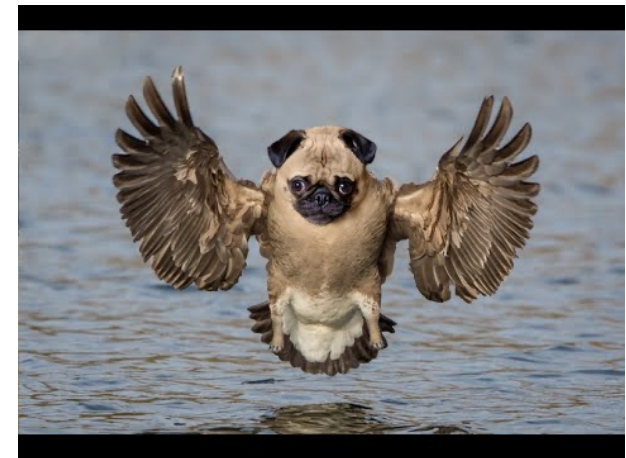
Contents

- ✓ Genetic algorithm
- ✓ Representations of chromo (integer, combinatorial)
- ✓ Selection strategies
- ✓ Fitness function
- ✓ Algorithm tuning, genetic drift



Contents

- ✓ Population management
- ✓ Another evolutionary algorithms
- ✓ Hybrid algorithms
- ✓ Multi-criteria algorithms, pareto-front
- ✓ Multipopulational algorithms, parallelization
- ✓ Coevolution algorithms



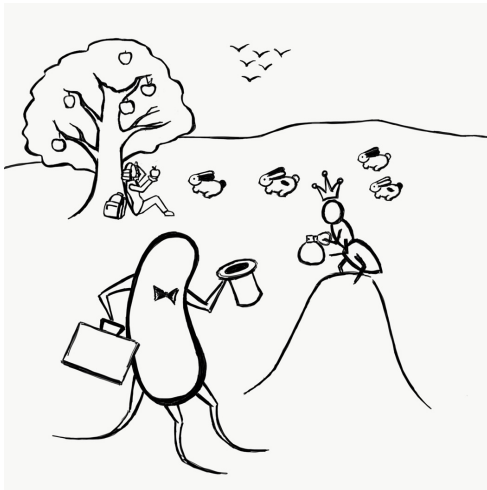
Practical tasks

- ✓ 1. Demonstrations
- ✓ 2. Real-valued optimization
- ✓ 3. Travelling salesman problem
- ✓ 4. Queens puzzle
- ✓ 5. Parallel models

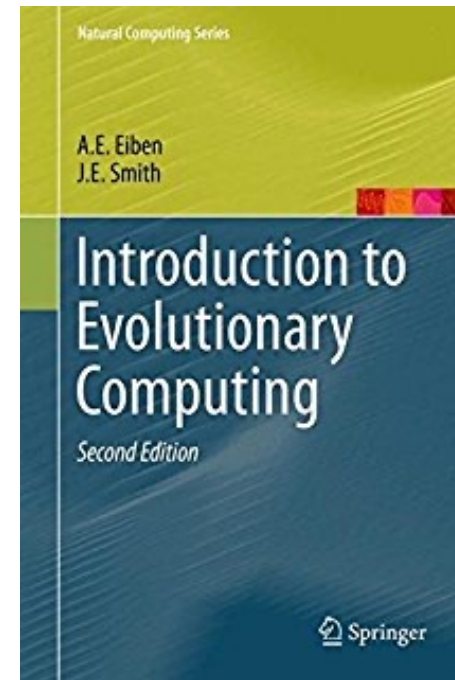
- ✓ Seminar classes:
 - Presentations on various population-based algorithms with discussion

Examples and literature

- ✓ <http://www.southampton.ac.uk/~ajk/truss/welcome.html>
- ✓ http://rednuht.org/genetic_walkers/
- ✓ http://rednuht.org/genetic_cars_2/
- ✓ <https://youtu.be/qv6UVOQ0F44>



<https://books.ifmo.ru/file/pdf/2389.pdf>



Introduction to Evolutionary Computing
Authors: **Eiben**, A.E., **Smith**, James E

Watchmaker demo

\\10.9.14.41\share\Evolutionary computing\soft\

Copy Watchmaker

Open IntelliJIdea -> import from Maven

✓ BitsExample.java

- Play around dimension of problem

✓ *MonaLisaApplet.java*

✓ TravellingSalesmanApplet.java

- Select all and play with parameters:
 - Population size – number of solutions in population
 - Elitism – number of best solutions to save



Thank you for your attention!

www.ifmo.ru

ITsMO *re than a*
UNIVERSITY