# Lecture 2

## Genetic algorithms

Genetic algorithms are based on the theory of natural selection. The set of solutions is called a **population**. Each solution in the population is equivalent to a **chromosome**. Solutions from one population are taken and used to form a new, potentially better population.

### Outline of the Basic Genetic Algorithm

1. **[Start]** Generate the initial (random) population of chromosomes.
2. **[Fitness]** Evaluate the fitness of each chromosome in the population.
3. **[New population]** New population is generated from the best individuals by repeating the following steps:
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population.
4. **[Replace]** Use the newly generated population for a further run of the algorithm.
5. **[Test]** Stop if the end condition is satisfied (result is the best solution in the current population)
6. **[Loop]** Go to step **2**

## Example

### Encoding of a chromosome as a binary string

| | |
|---|---|
| Chromosome 1 | 1101100100110110 |
| Chromosome 2 | 1101111000011110 |

**Crossover**

Choose randomly a crossover point. Copy from a first parent everything before this point. Copy from the second parent everything after a crossover point.

| | |
|---|---|
| Chromosome 1 | 11011 \| 00100110110 |
| Chromosome 2 | 11011 \| 11000011110 |
| Offspring 1 | 11011 \| 11000011110 |
| Offspring 2 | 11011 \| 00100110110 |

**Mutation**

Mutation changes randomly the new offspring, e.g. by switching with some mutation probability bits from 1 to 0 or from 0 to 1. If mutation probability is 100%, the whole chromosome is changed, if it is 0%, nothing is changed.

| | |
|---|---|
| Original offspring 1 | 1101111000011110 |
| Original offspring 2 | 1101100100110110 |
| Mutated offspring 1 | 1100111000011110 |
| Mutated offspring 2 | 1101101100110110 |

**Elitism**

Method, which first copies the given number of best chromosomes to the new population.

## Task 1. Knapsack optimization model (binary chromosome)

(source: *https://www.r-bloggers.com/2012/08/genetic-algorithms-a-simple-r-example*)

You are going to spend a month in the wilderness. You're taking a backpack with you, however, the maximum weight it can carry is 20 kilograms. You have a number of survival items available, each with its own number of "survival points". You're objective is to maximize the number of survival points.

The following table shows the items you can choose from.

| item | survivalpoints | weight |
|---|---|---|
| pocketknife | 10.00 | 1.00 |
| beans | 20.00 | 5.00 |
| potatoes | 15.00 | 10.00 |
| unions | 2.00 | 1.00 |
| sleeping bag | 30.00 | 7.00 |
| rope | 10.00 | 5.00 |
| compass | 30.00 | 1.00 |

## Task 2. Optimisation of stores supplies (floats chromosome)

There are three producers of the same product, located in Łódź, Poznań and Warsaw. The recipients of this product are stores located near five large cities (Gdańsk, Olsztyn, Kraków, Wrocław, Katowice).

Producers have a predetermined production capacity, so a production volume ceiling cannot be exceeded. It is respectively:

- Łódź: 750 units,
- Poznań: 640 units,
- Warsaw: 450 units.

Stores report a specific demand for a product. This amount determines the minimum number of product units delivered to a given store: it can be greater but not less than the reported demand. The values of the demand for the product in individual stores are as follows:

| Store | Demand |
|---|---|
| Gdańsk | 180 |
| Olsztyn | 80 |
| Kraków | 200 |
| Wrocław | 160 |
| Katowice | 220 |

Transport costs varies depending on the manufacturer from which the product is delivered to the store. Distribution costs (**USD per one unit**) of products between producers and stores are presented in the table below:

| Store \ Producer | Łódź | Poznań | Warszawa |
|---|---|---|---|
| **Gdańsk** | 100 | 60 | 30 |
| **Olsztyn** | 80 | 50 | 40 |
| **Kraków** | 60 | 40 | 50 |
| **Wrocław** | 50 | 30 | 50 |
| **Katowice** | 40 | 60 | 90 |

The problem is defining a product delivery plan from a specific producer to a particular store to minimise the total transport costs. **We assume that units are divisible.**

## Task 3. Optimisation of stores supplies (integer chromosome)

Transport costs (for one truck) varies depending on the manufacturer from which the product is delivered to the store. The number of necessary trucks is calculated assuming that all trucks are identical and each truck carries a maximum of six units of product. Distribution costs (**USD per one truck**) of products between producers and stores are presented in the table above.

The problem is defining a product delivery plan from a specific producer to a particular store to minimise the total transport costs. **We assume that units are not divisible.**