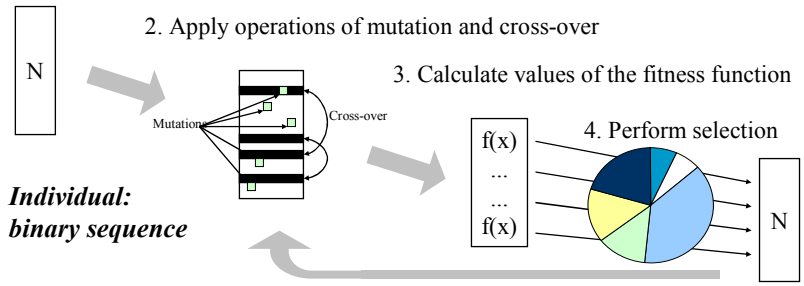
Evolutionary algorithms (2)

GENETIC ALGORITHM

Goal: Find maximum of function f(x)
Assumption: function is positive

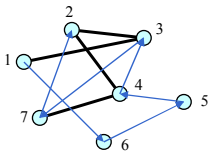
1. Generate N individuals randomly



5. Repeat from 2.

ENCODING OF PERMUTATIONS

In some applications, the binary encoding is less intuitive than the other possibilities. For example, in case of the Travelling Salesman Problem, it is easier to encode the routes as permutations.



Individual:
{ 1 6 5 4 3 7 2 }

Genetic algorithm can work with individuals-permutations. The only problem is to redefine genetic operators (mutation and cross-over): one must enable them to result always with valid permutations

GENETIC OPERATORS

Mutation: transposition of two randomly chosen positions

{ 1 4 2 6 3 0 7 } ⟶ { 1 4 7 6 3 0 2 }

Cross-over OX (*order cross-over*): Choose randomly the intersection position for the given two chromosomes. Keep the parts before this position unchanged. Values after the chosen position should be sorted with respect to the order of their appearance in the other individual.

{ 1 4 0 5 | 7 3 2 6 }
{ 4 0 7 6 | 2 3 5 1 }
⟶
{ 1 4 0 5 7 6 2 3 }
{ 4 0 7 6 1 5 3 2 }

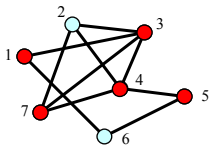
This is just an example: various cross-over operators are used (OX is one of the easiest approaches)

EXAMPLE

Heuristic approach to searching for the Minimal Edge Covering within the given graph: Begin with a randomly chosen ordering over vertices. Then, keep adding them one by one to the covering being constructed, unless a given vertex does not cover any extra edge with respect to the previously considered vertices.

Modification: Instead of choosing the ordering randomly, apply genetic algorithm working with permutations.

Individual:
{ 1 5 6 4 3 7 2 }



Genenetic → ordering of vertices → Heuristic → covering

ITERATED PRISONER DILEMMA

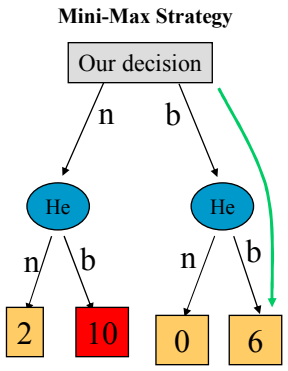
- We have two prisoners (in separate cells), who participated in the bank robbery.
- Each of them has two possibilities: **betray** or **not**
  - If Prisoner A **betrays** Prisoner B, then A is free and B gets 10 years (and vice versa)
  - If they both **don't betray**, then they get 2 years
  - If they both **betray**, then they get 6 years
- How Prisoner A should behave?

Iterated version: *A meets B in prison again and again*
Hence, A remembers B's behavior in the past

STRATEGY

Strategy: the ensemble of laws (algorithm) explaining how to behave in an arbitrary situation, which may occur during the game

In practice: the function with the history of previous cases as input and the answer „betray or not" as output



We behave in purpose of minimizing the loss which can caused by our opponent

ENCODING OF STRATEGY

- Assume that two previous „games" are stored in memory as two pairs consisting of two actions (e.g. „bn" – I betrayed and he didn't)
  - 16 possibilities for such pairs (e.g. „bnbb")
  - 4 additional possibilities in case of the 2nd step
  - 1 additional possibility in case of the 1st step
- **Strategy:** 21 rules, e.g., bnbb=>b, nn=>n, b... can be encoded by 21 bits (the right sides) – it is enough to find optimal 21-bit sequence
- **Fitness:** the tournament within population ("everyone vs. everyone") – the value of the fitness function is the sum of obtained points

GENETIC PROGRAMMING

Exemplary task:
Write an algorithm for a machine equipped with sensors and output devices, in purpose of getting through a labyrinth. Machine can additionally use internal memory to keep helpful information.

- Input: the state of registries and sensors (For instance: Is there an obstacle ahead?)
- Output: the state of output devices (For instance: Go ahead, Turn to the right, Change the state of registries).

ENCODING OF ALGORITHMS

The robot's programming language consists of:
- Logical operation: IF(condition,action1,action2)
- Grouping operations: GROUP3(action1,action2,action3)
- Output operations: STEP, LEFT, RIGHT, SET(registry,value),
- Predicates denoting the state of inputs and registries: WALL, WALL_L, WALL_R, ...

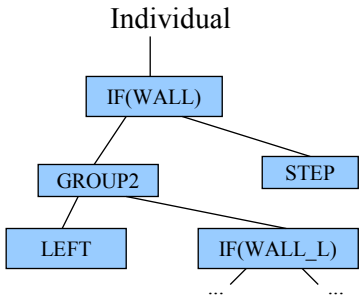
The goal of the genetic algorithm is to create a program – the tree of combinations of admissible operations and conditions
Evaluation of an individual (program) should base on simulation of its performance in the testing environment

ENCODING OF ALGORITHMS

Individual: program written by means of the tree structure. The leaves correspond to commands with no arguments

Mutations: random changes within the content of a node, exchange of a leaf on a node with children, deleting a fragment of the tree...

Cross-over: random exchange of nodes with corresponding subtrees



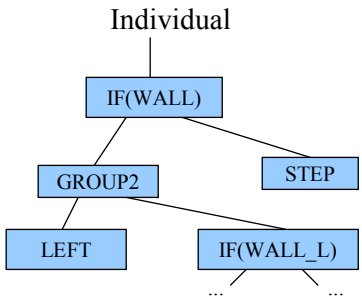
Cross-over: exchange of subtrees

Be careful about the tree growth

ENCODING OF ALGORITHMS

Genetic programming is usually connected with simulation of corresponding programs and evaluation of the results of their performance. This is a very time-consuming process. Typical size of population is equal to several thousands of individuals.

Calculation of the fitness function: Generate labyrinth, put the machine inside, simulate, e.g., 1000 steps of the program according to the given tree, calculate the distance to the exit from the labyrinth



APPLICATIONS (2000)

- Optimization of engines in B777
- Optimization algorithms for planning of the agricultural tasks
- Stock market – average sum of transactions performed by basing on genetic algorithms is estimated as 28 billions USD a day
- Police – programs fitting faces on pictures to the specified persons