

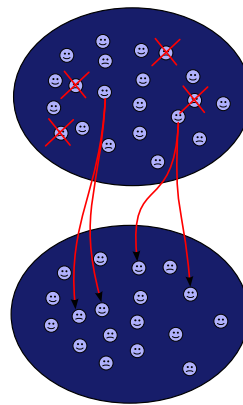
Genetic Algorithms

- 1 Foundations
- 2 Algorithm Components
 - Coding of Hypotheses
 - Fitness Functions
 - Selection
 - Variation
- 3 Numerical Optimization
- 4 Genetic Programming
 - Example

Genetic Algorithms

Parallel optimization inspired by biological evolution

- Populations of Hypotheses
- Selection Process
- Local Variation



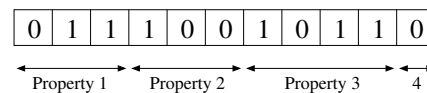
- Population of **Individuals**
- **Selection** of the best individuals
- **Variation** creates new individuals
- New **Generations** created iteratively

- 1 Foundations
- 2 Algorithm Components
 - Coding of Hypotheses
 - Fitness Functions
 - Selection
 - Variation
- 3 Numerical Optimization
- 4 Genetic Programming
 - Example

Coding of Hypotheses

How are different hypotheses stored?

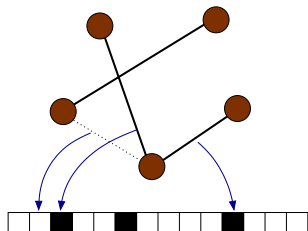
Chromosomes — Binary Strings



- Genotype
The actual representation (the chromosome)
- Fenotype
Properties of the individual (interpretation)

Example: Optimal choice of edges in a graph

The edges are encoded as a bit string



Do we *have* to use bit strings?

Variants:

- Other integers than only 0/1
- Real numbers
- Variable length
- Tree structures

Fitness Function

Measure of how good the hypothesis is

$$f : \text{chromosome} \mapsto \mathbb{R}$$

Example:

- Total path length in a graph
- Error in a function approximation
- Performance of a simulated robot
- Number of games won

Evaluating the fitness functions is normally the *most time consuming* part of a genetic algorithm

Selection

Basic idea: Preserve individuals with a high fitness

- **Roulette selection**
Probability of survival proportional to f
- **Ranking selection**
Selection based on order instead of the actual fitness value
- **Tournament selection**
Random pairs are formed and the one with highest fitness survives
- **Elitism**
The best individuals in a generation are guaranteed to survive

- **Mutations**
Small random modifications
- **Crossovers**
Mixing of individuals content

Mutations

- Make random changes to the contents of the chromosome
- Choice of coding makes a big difference

Variation

Crossovers

- Select two individuals with high fitness
- Exchange parts of the chromosome with each other

One-point crossover
Multi-point crossover

Application on ordinary optimization problems

Assume that we are looking for $\max f(x, y)$

Encoding: chromosome consisting of two real numbers

Each individual corresponds to a point in the plane

- Mutations
Redistribution parallel to the x and y axis
- Crossovers
New points with x from one parent and y from the other

Example: Optimized code generation from a compiler

ACOVEA — Analysis of Compiler Options via Evolutionary Algorithms

Software for finding the optimal compiler options for a given C program

Genetic Programming

The use of GA to automatically create programs

- How are programs represented?
- How can one measure fitness?
- How are mutations done?
- How are crossovers done?

Representation of Programs

Ordinary programming languages are not suitable

- Tree with operators
- List of instructions

Example

Function Approximation

Representation of the program

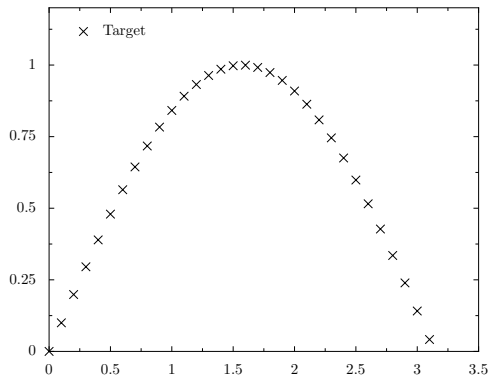
```

graph TD
    A((+)) --> B((*))
    A --> C((abs))
    B --> D((17))
    B --> E((X))
    C --> F((X))
    
```

- Mutations
- Crossovers

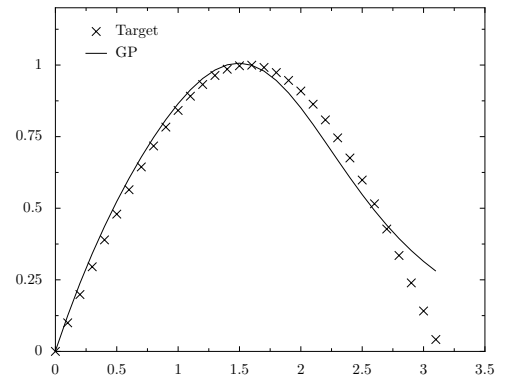
Example

Goal Function

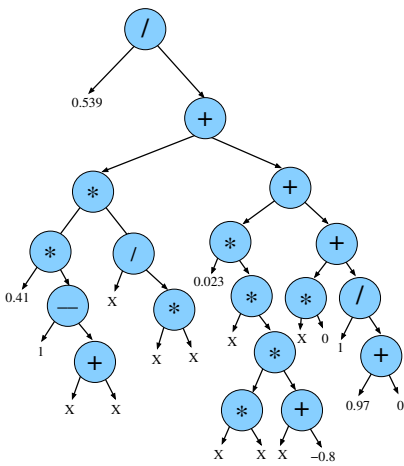


Example

Solution found by the algorithm



Example



Bloating
 Accumulation of unnecessary parts in chromosomes with variable length