HØGSKOLEN I ÅLESUND

Ingeniør

GENETIC ALGORITHMS: AN INTRODUCTION

Date: Tuesday 17 February 2015

Course: Functional Programming and Intelligent Algorithms

Lecturer: Robin T. Bye

Natural optimisation methods

Nature-inspired algorithms

- Genetic algorithm (GA)
- Genetic programming (GP)
- Simulated annealing (SA)
- Particle swarm optimisation (PSO)
- Ant colony optimisation (ACO)
- Evolutionary strategies (ES)
- Cultural algorithms (CA)
- +++

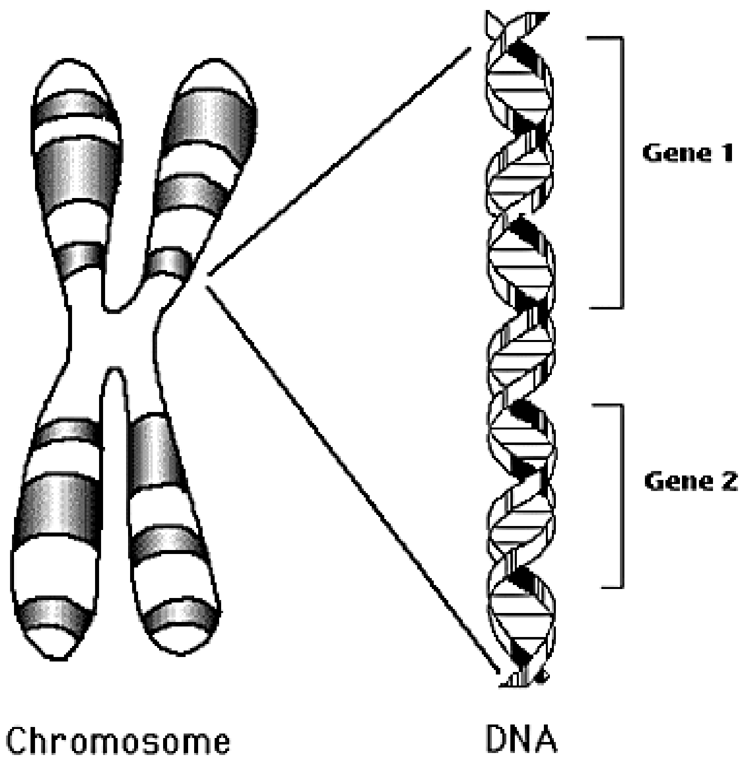
Natural selection

- Biological optimisation
- Observations from nature:
  1. Extreme *diversity* of organisms
  2. Large degree of *complexity* in organisms
  3. Organism features with great *usefulness*

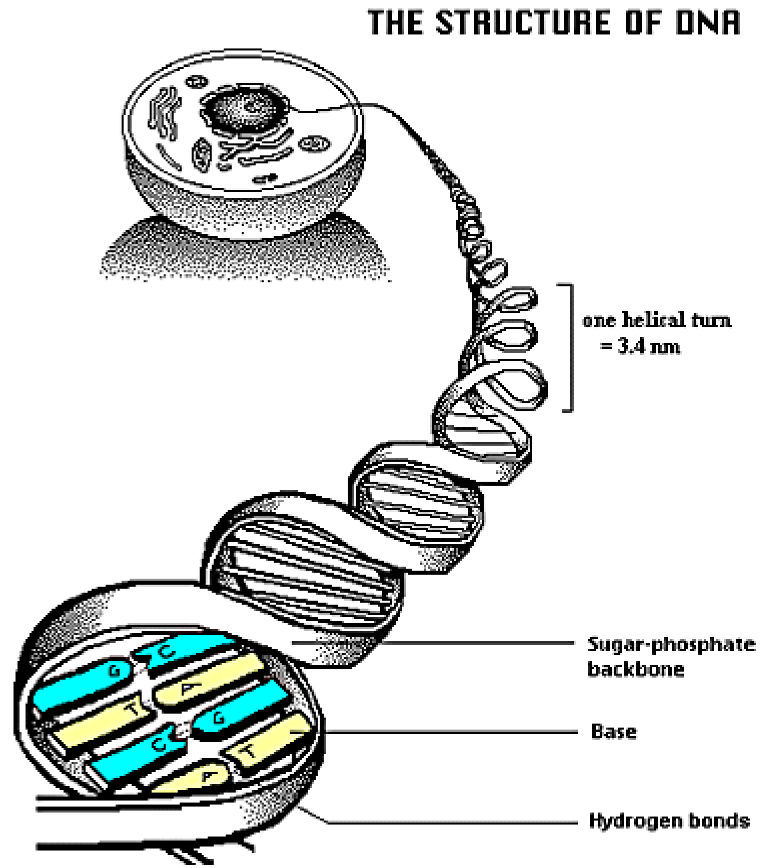
  → Survival of the fittest
- Natural selection have two components:
  - Genetics
  - Evolution

Genetics

- Gene: Basic unit of heredity
  - Determines characteristics or functions of an individual
- Chromosome: Set of genes
- DNA: Set of chromosomes
- Humans: 1 DNA with 23 pairs (=46) of chromosomes with about 40 000 genes


Gene 1
Gene 2
Chromosome
DNA

Genes


THE STRUCTURE OF DNA
one helical turn = 3.4 nm
Sugar-phosphate backbone
Base
Hydrogen bonds

Evolution

- Darwin's four premises:
    1. Offspring inherits many characteristics from parents
    2. Variations in characteristics between individuals can be passed on
    3. Only a small percentage of offspring survives to adulthood
    4. Survival is dependent on inherited characteristics

ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE
FOR LIFE.

BY CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE
ROUND THE WORLD.'

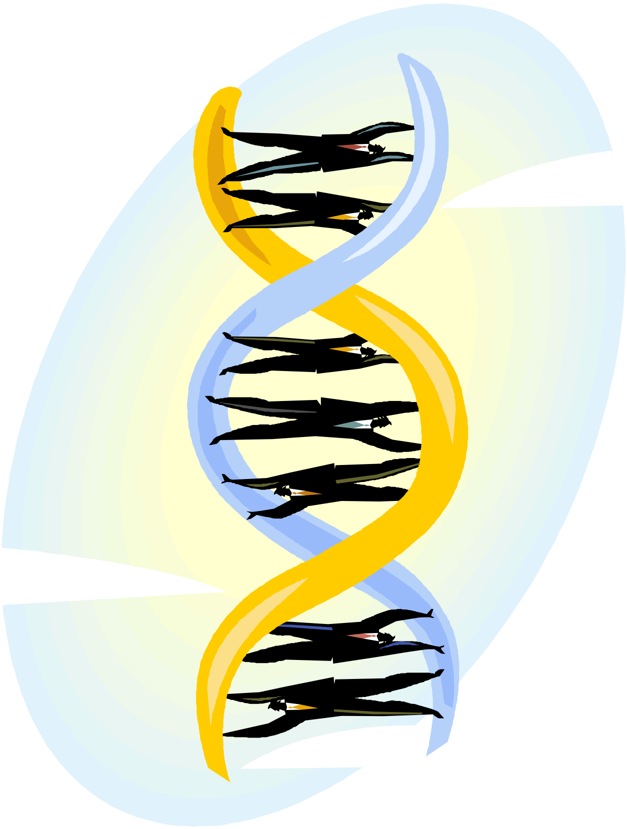
Evolution

- Static population –> No evolution
- Dynamic population requires
  - Mutations (random changes in gene characteristics)
  - Gene flow (new organisms join the population)
  - Genetic drift (occurs due to chance)
  - Natural selection (survival of the fittest)

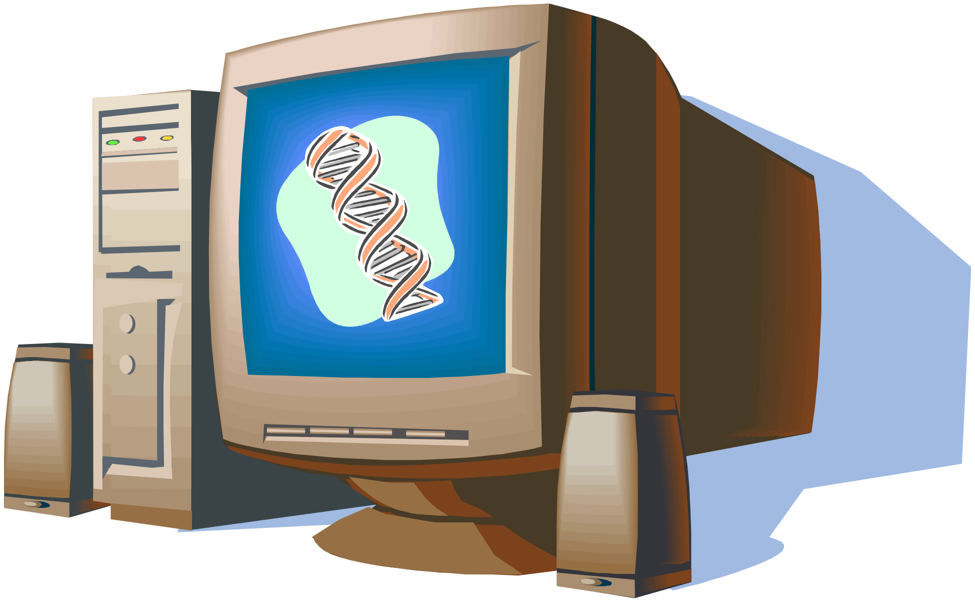
Simulation of natural evolution

- Evolutionary computing is based on:
  - Creating an initial population
  - Evaluate the fitness of each individual
  - Generate new population through genetic operations
  - Repeat until desired fitness reached

Natural evolution



simulated evolution



Neo-Darwinian paradigm

- Mendel: Genetics
- Darwin, Wallace: Theory of evolution
- Weismann: Theory of natural selection
- Together → Neo-Darwinism
- Central processes:
  - Reproduction, mutation, competition, selection

Evolutionary fitness

- A population's ability to survive and reproduce over several generations
- Ability to anticipate environmental changes
- Is optimised in natural life

The genetic algorithm (GA)

Principles of the GA

- Optimisation and search technique based on principles of genetics and natural selection
- A population with many individuals evolves under specified selection rules
- Each generation improves fitness (minimises cost function) until satisfied

History

- Developed by John Holland (1970s)
- Popularised by David Goldberg (1990s)
  - Holland’s aim: Imitate nature’s evolution with computers
  - Concerned with bit strings
  - 1 bit = 1 artificial gene (either 0 or 1)
  - Bit string = artificial chromosome

1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	1

Encoding and evaluation

- Encoding and evaluation mechanisms link GA to the problem to be solved
- Encoding:
  - How to represent candidate solutions
  - Binary, continuous-valued, other schemes
- Evaluation:
  - How good is a candidate solution?
  - Choice of cost function, performance of chromosome (candidate solution)

Advantages of GA

- Can optimise with both discrete and continuous variables
- Does not need derivative information
- Simultaneously searches a wide sampling of the cost surface
- Handles large number of variables
- Well suited for parallel computing
- Can optimise in complex cost surfaces

Advantages of GA

- Does not get stuck in local minima
- Can provide several optimal solution candidates, not just one solution
- Can encode variables so optimisation is done with encoded variables
- Works with
  - numerically generated data
  - experimental data
  - analytical functions
- Can stop GA anytime with a (working) non-optimal solution that is often good enough

Basic steps in a GA

1. Represent the problem variable domain (encoding)
2. Define a cost function
3. Generate initial population
4. Calculate cost of each chromosome
5. Select parent chromosomes for mating based on cost
6. Create offspring chromosome based on genetic operators crossover and mutation
7. Place offspring in new population
8. Repeat Steps 5-7 until size of new population equals initial population
9. Replace initial population (parents) with new population (offspring)
10. Go to Step 4 and repeat steps until stopping criteria (reached global minimum, certain cost, max number of iterations, or other)

Reproduction

- Select some chromosomes in population for reproduction, eg.,
  - top half of a ranked list
  - roulette wheel selection
- Use genetic operators for reproduction:
  - Crossover
  - Mutation

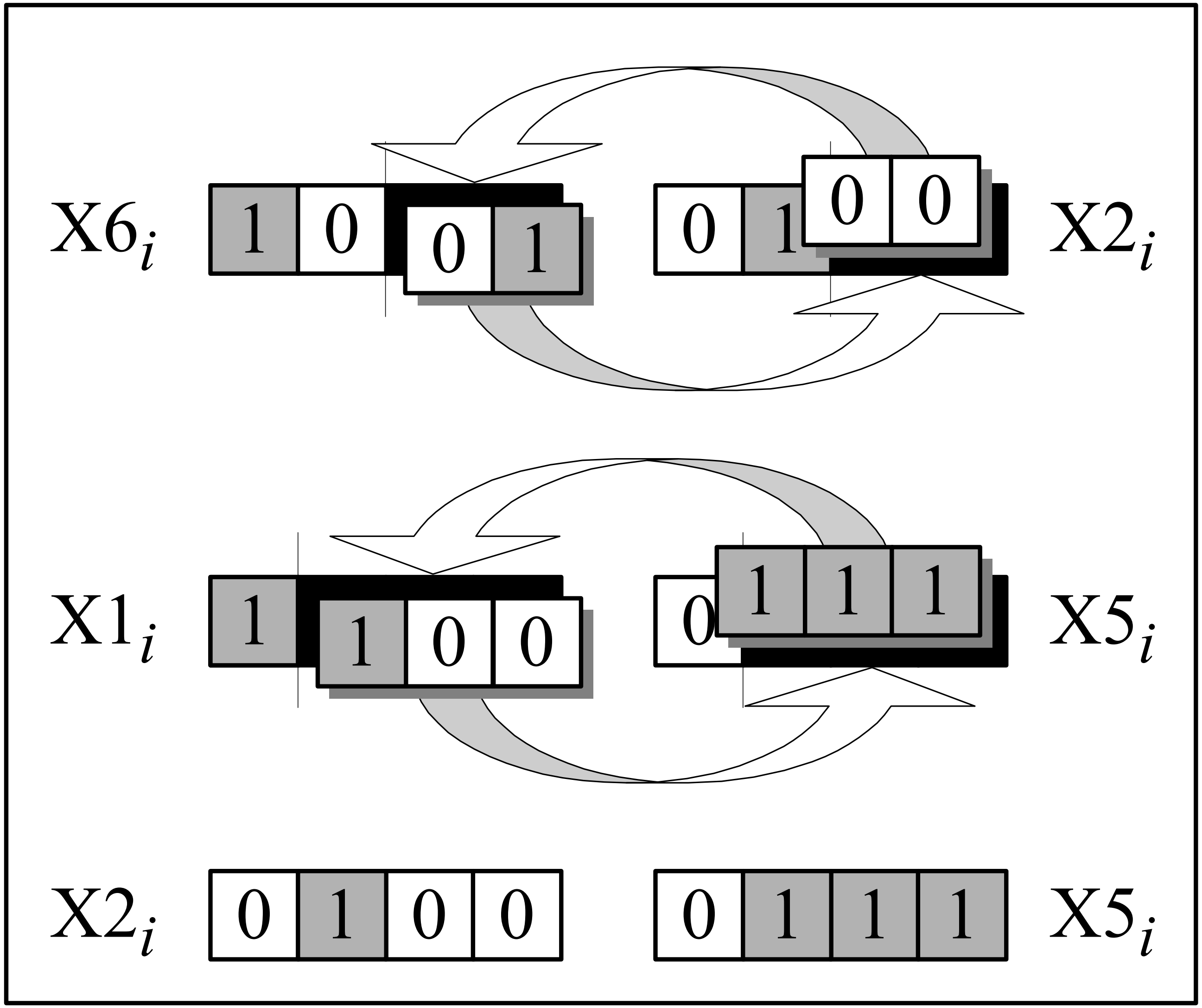
Crossover

- Two parent chromosomes are each split up in two parts at crossover point
- Swap all genes to the right of crossover point to create offspring chromosomes
- Let crossover only happen with some probability
- No crossover → clone parents

Crossover at 3rd gene

Crossover at 2nd gene

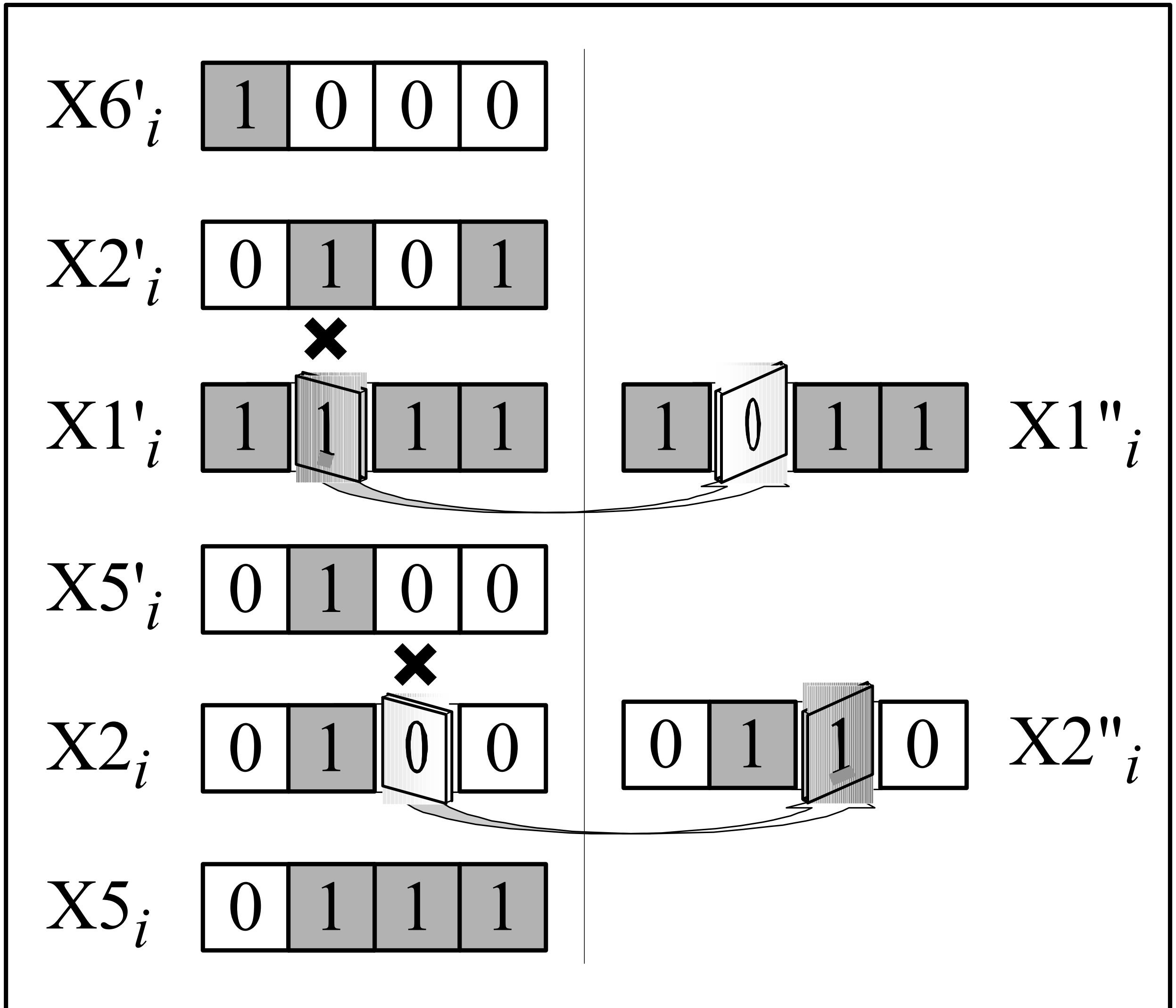
Cloning



Task: Make a table showing chromosomes before and after

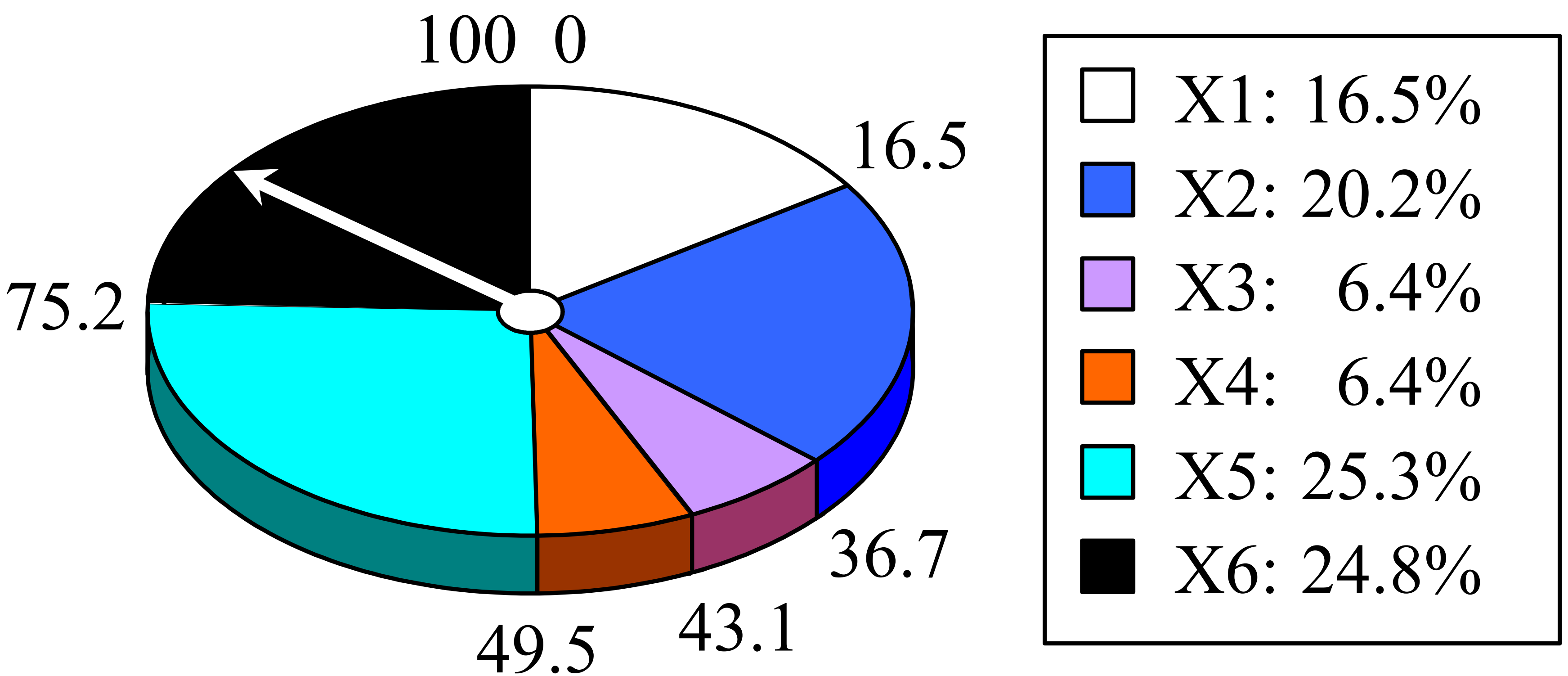
Mutation

- Change in a gene, e.g. flip a 1 to a 0
- Randomly select which gene
- Some small probability of mutation to occur
- Aids against getting trapped in local minimum

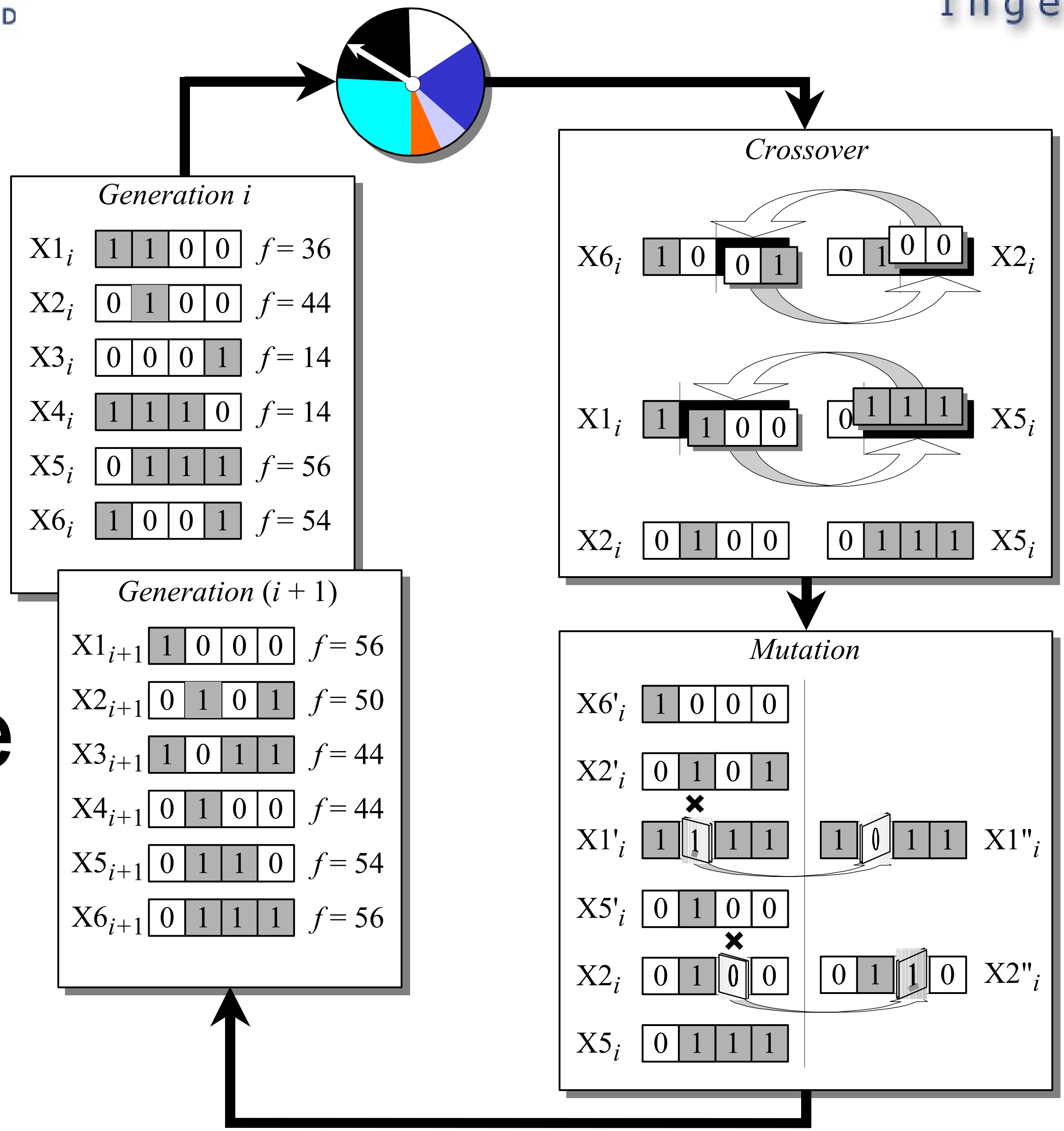

X6'i
1 0 0 0
X2'i
0 1 0 1
X1'i
1 1 1 1
1 0 1 1
X1"i
X5'i
0 1 0 0
X2i
0 1 0 0
0 1 1 0
X2"i
X5i
0 1 1 1

Roulette wheel selection

- Rank chromosomes in list
- Give a weight according to fitness/ability to minimise cost
  - High fitness, high weight
- Spin wheel and randomly draw chromosomes
- Fit chromosomes have higher probabiliy of getting picked


100
0
16.5
36.7
43.1
49.5
75.2
X1: 16.5%
X2: 20.2%
X3: 6.4%
X4: 6.4%
X5: 25.3%
X6: 24.8%

GA cycle


Generation i
X1i 1 1 0 0 f = 36
X2i 0 1 0 0 f = 44
X3i 0 0 0 1 f = 14
X4i 1 1 1 0 f = 14
X5i 0 1 1 1 f = 56
X6i 1 0 0 1 f = 54
Generation (i + 1)
X1i+1 1 0 0 0 f = 56
X2i+1 0 1 0 1 f = 50
X3i+1 1 0 1 1 f = 44
X4i+1 0 1 0 0 f = 44
X5i+1 0 1 1 0 f = 54
X6i+1 0 1 1 1 f = 56
Crossover
X6i 1 0 0 1
0 1 0 0 X2i
X1i 1 1 0 0
0 1 1 1 X5i
X2i 0 1 0 0
0 1 1 1 X5i
Mutation
X6'i 1 0 0 0
X2'i 0 1 0 1
X1'i 1 1 1 1
1 0 1 1 X1"i
X5'i 0 1 0 0
X2i 0 1 0 0
0 1 1 0 X2"i
X5i 0 1 1 1

Other selection methods

- Tournament selection: Picks a subset of the population at random, then selects the members with the best fitness.
- Truncation selection: Chromosomes whose cost is below a certain value (truncation point) are selected as parents for the next generation.

Other selection methods

- Single-point crossover: A random-point in two chromosomes is selected. Offspring are formed by joining the genetic material to the right of the crossover point of one parent with the genetic material to the left of the crossover point of the other parent.

Other selection methods

- Two-point crossover: Choose two points in the chromosome and exchange bits between those two points.
- Uniform crossover: Randomly assign the bit from one parent to one offspring and the bit from the other parent to the other offspring.