LUT University

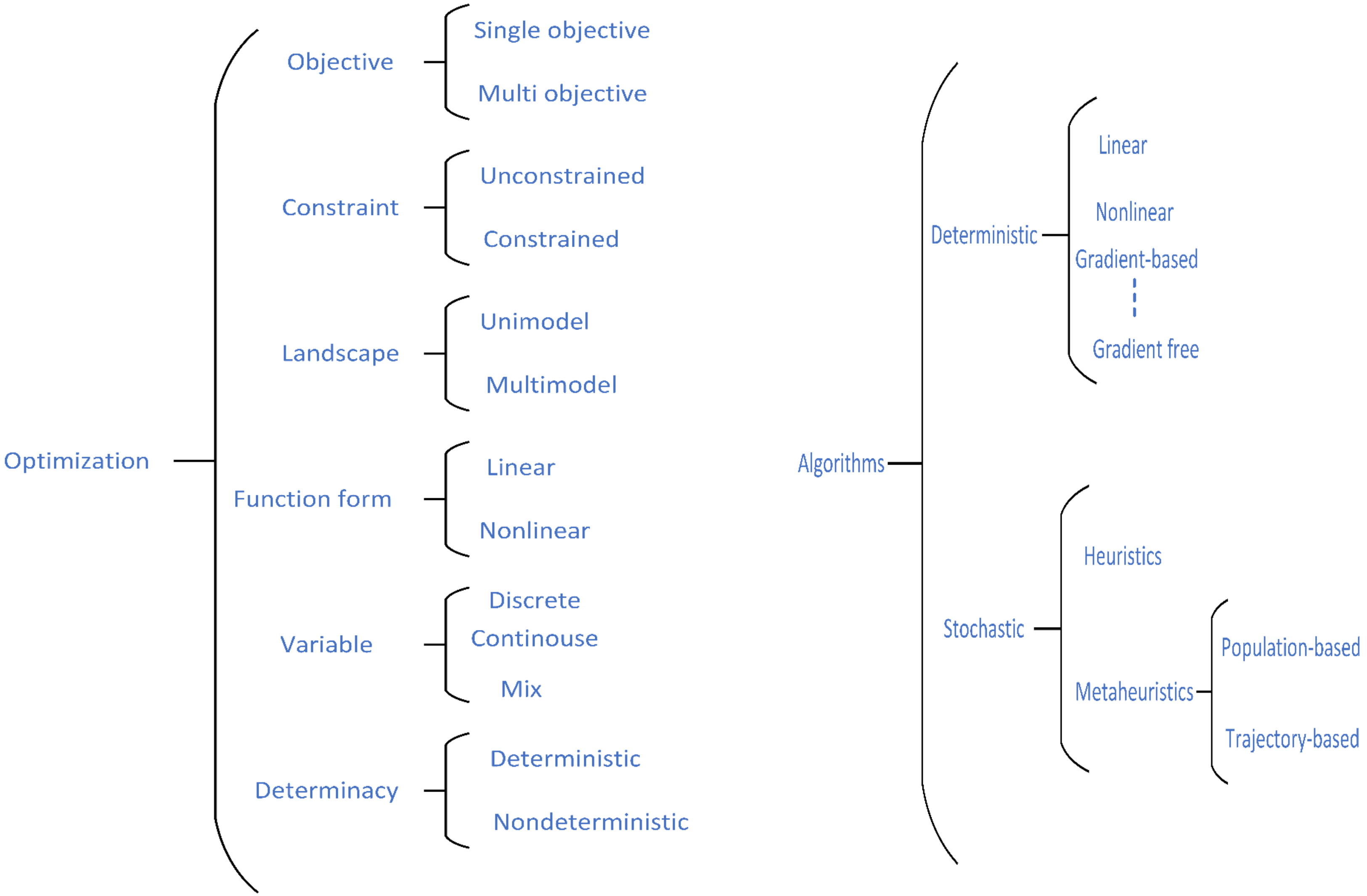
Introduction to Genetic Algorithm (GA)

Introduction to optimization (2/13)

- Optimization can include a wide range of problems with the aim of searching for certain optimality. Subsequently, there are many different ways of naming and classifying optimization problems, and typically the optimization techniques can also vary significantly from problem to problem.

- Generally speaking, classification can be carried out in terms of the number of objectives, number of constraints, function forms, landscape of the objective functions, type of design variables, uncertainty in values, and computational effort.

Introduction to optimization (3/13)



- Optimization
  - Objective
    - Single objective
    - Multi objective
  - Constraint
    - Unconstrained
    - Constrained
  - Landscape
    - Unimodel
    - Multimodel
  - Function form
    - Linear
    - Nonlinear
  - Variable
    - Discrete
    - Continouse
    - Mix
  - Determinacy
    - Deterministic
    - Nondeterministic

- Algorithms
  - Deterministic
    - Linear
    - Nonlinear
    - Gradient-based
    - Gradient free
  - Stochastic
    - Heuristics
    - Metaheuristics
      - Population-based
      - Trajectory-based

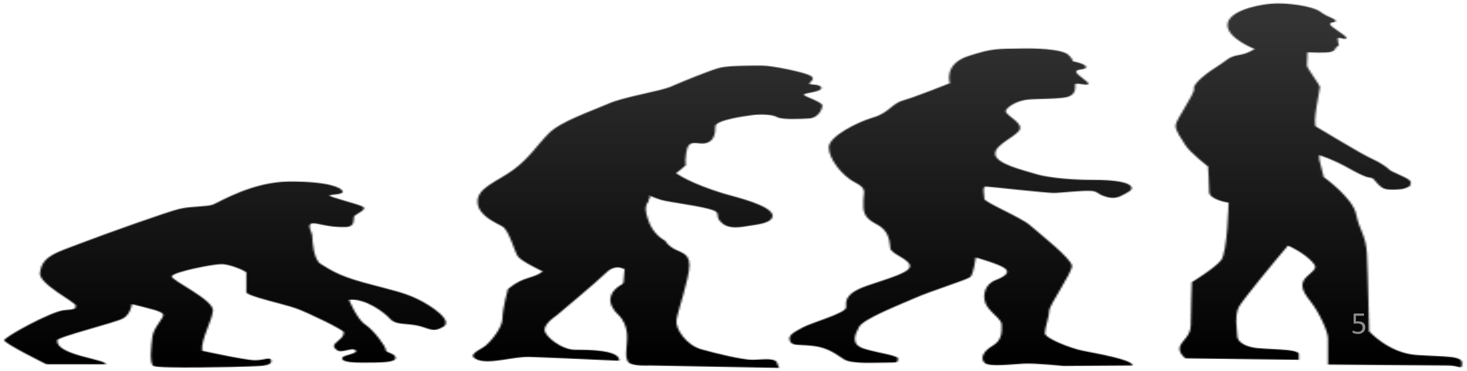
GA (4/13)

- **Introduction**
  - Based on Darwin’s theory of evolution
  - Rapidly growing area of artificial intelligence
  - Used to solve optimization-based problems
  - Techniques inspired by evolutionary biology
    - Inheritance
    - Mutation
    - Selection
    - Crossover
- **History**
  - Evolutionary computing evolved in the 1960’s by I.Rechenberg
  - GA was invented by John Holland in the mid-70’s.

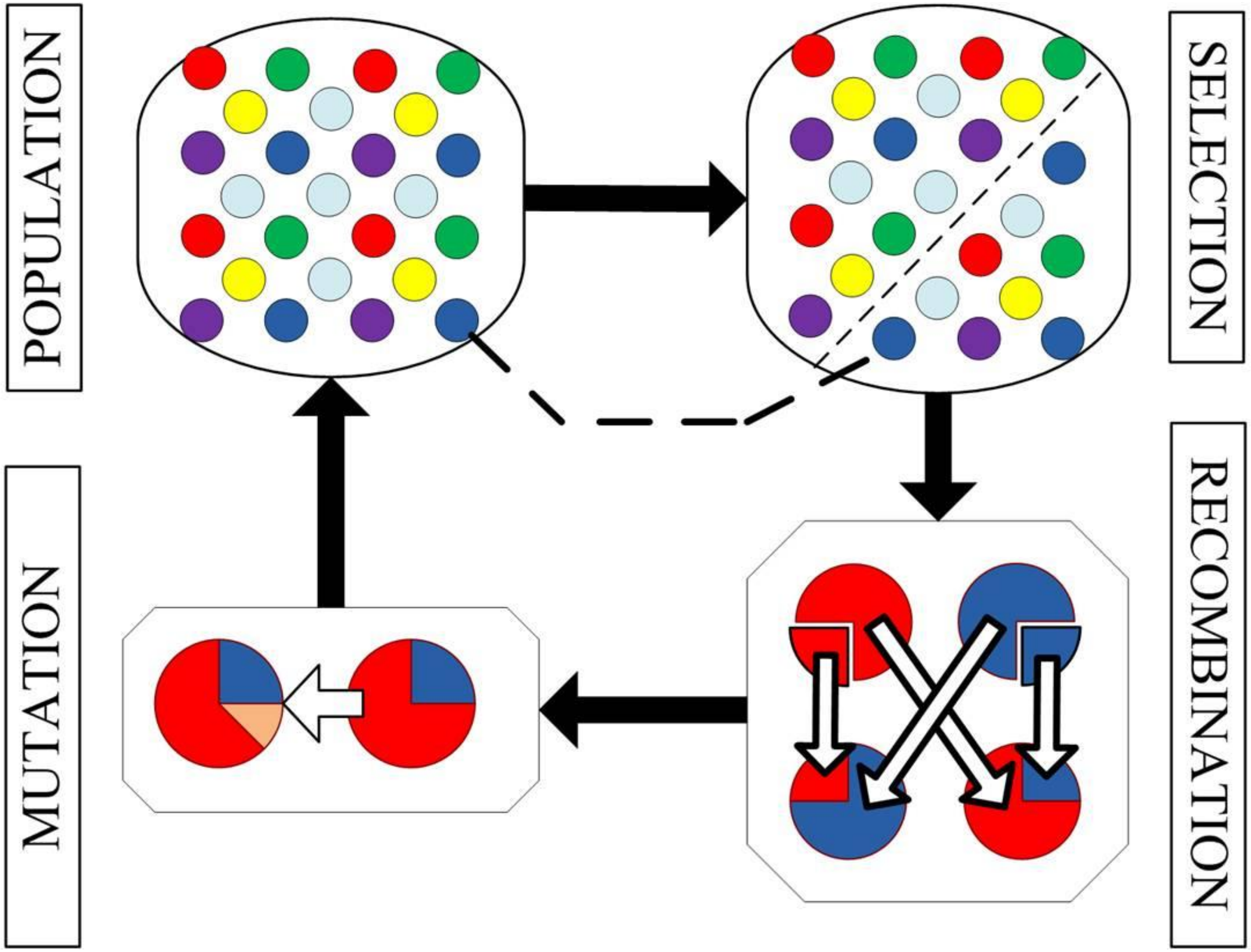
GA (5/13)

- **Natural selection**
  - Only the organisms best adapted to their environment tend to survive
  - Transmit their genetic characteristics in increasing numbers to succeeding generations
  - Those less adapted tend to be eliminated

- **GA is inspired from nature**
  - A genetic algorithm maintains
    - Population of candidate solutions for the problem
    - Makes it evolve by iteratively applying a set of stochastic operators



GA (6/13)



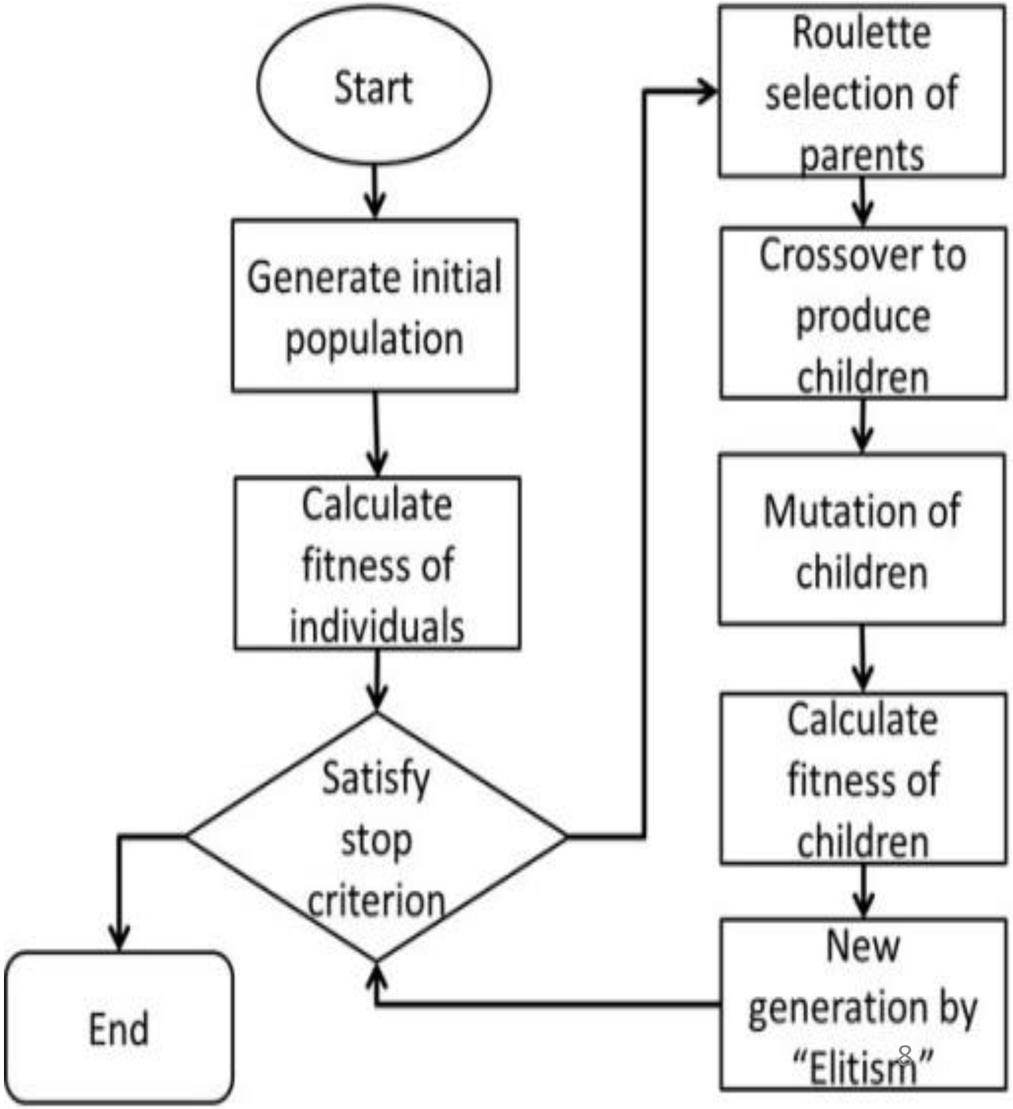
GA (7/13)

- **GA requirements**
  - A genetic representation of the solution domain (set search space)
  - A fitness function to evaluate the solution domain

  - **Representation**
  - Chromosomes could be
    - Bit strings (0101 ... 1100)
    - Real numbers (43.2 -33.1 ... 0.0 89.2)
    - Permutations of element (E11 E3 E7 ... E1 E15)
    - Lists of rules (R1 R2 R3 ... R22 R23)
    - Program elements (genetic programming)
    - ... any data structure ...

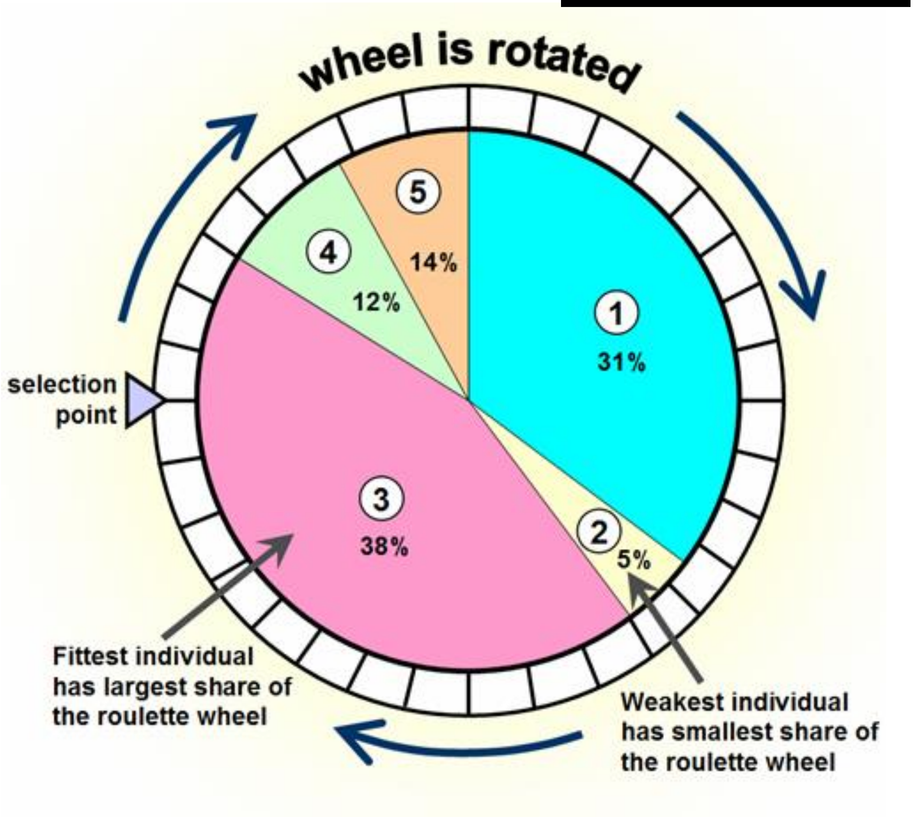
GA (8/13)

- **Algorithm**
  1. Initialization
  2. Generate population
  3. Evaluate fitness
  4. **While** gen<max_gen **do**
  5. Select two parents
  6. Crossover
  7. Mutation
  8. Evaluate fitness
  9. Elitism
  10. **End**


Start
Generate initial population
Calculate fitness of individuals
Satisfy stop criterion
End
Roulette selection of parents
Crossover to produce children
Mutation of children
Calculate fitness of children
New generation by "Elitism"

GA (10/13)

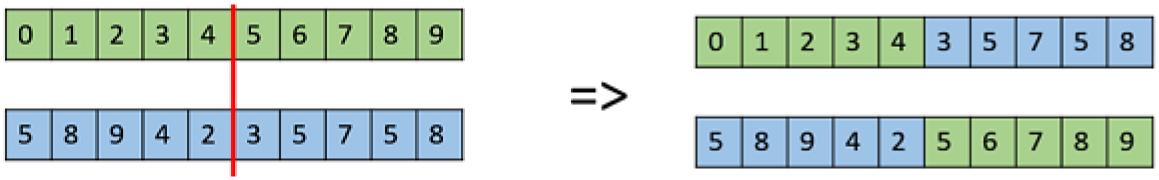
- **Selection**
  - Darwinian survival of fittest
  - More preference to get better guys
  - Ways to select
    - Roulette wheel
    - Tournament
    - Truncation
  - By itself, pick best


wheel is rotated
5
14%
4
12%
1
31%
selection point
3
38%
2
5%
Fittest individual has largest share of the roulette wheel
Weakest individual has smallest share of the roulette wheel

- **Roulette wheel**
  - Probability of selection is assigned to individuals based on their fitness value
  - Two parents are selected randomly from set of individuals with high selection probability
  - Selected parents are used to produce off-springs

GA (11/13)

- **Crossover**
  - Single point crossover



0 1 2 3 4 5 6 7 8 9

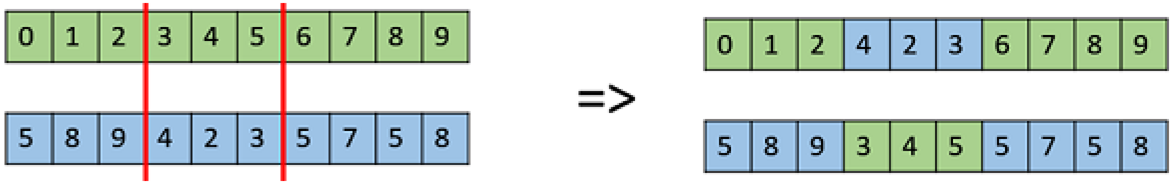
5 8 9 4 2 3 5 7 5 8

=>

0 1 2 3 4 3 5 7 5 8

5 8 9 4 2 5 6 7 8 9

  - Two point crossover



0 1 2 3 4 5 6 7 8 9

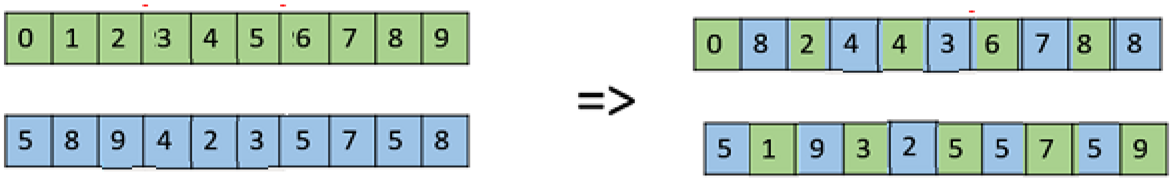
5 8 9 4 2 3 5 7 5 8

=>

0 1 2 4 2 3 6 7 8 9

5 8 9 3 4 5 5 7 5 8

  - Uniform crossover



0 1 2 3 4 5 6 7 8 9

5 8 9 4 2 3 5 7 5 8

=>

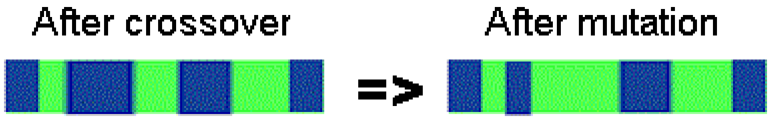
0 8 2 4 4 3 6 7 8 8

5 1 9 3 2 5 5 7 5 9

GA (12/13)

- **Mutation**
  - String subjected to mutation after crossover
  - It prevents falling all solutions into local optimum
  - It alters bits of chromosomes to make the string better
  - Bit wise mutation is performed with probability of mutation Po
  - A random number ro is generated and compared with Po
  - If Po>ro , then mutation occurs

- **Types of mutation**
  - ○ Bit inversion
  - ○ Selected bits are inverted



11001001 => 10001001

  - Order changing
    - ○ The number are selected and exchanged

(1 2 3 4 5 6 8 9 7) => (1 8 3 4 5 6 2 9 7)

GA (13/13)

- **Advantages of GA**
  - Concepts are easy to understand
  - Always an answer; answer gets better with time
  - Chances of getting optimal solution
- **Limitations**
  - The population considered for the evolution should be moderate or suitable
  - Accurate tuning of stochastic operators

Questions

Email: Majid.Hussain@lut.fi

Pseudocode: https://github.com/HM-colb/Heuristic-algorithm