Universidade do Minho
Escola de Engenharia
Departamento de Informática

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Computação Natural
2020/2021

Filipe Gonçalves, Paulo Novais

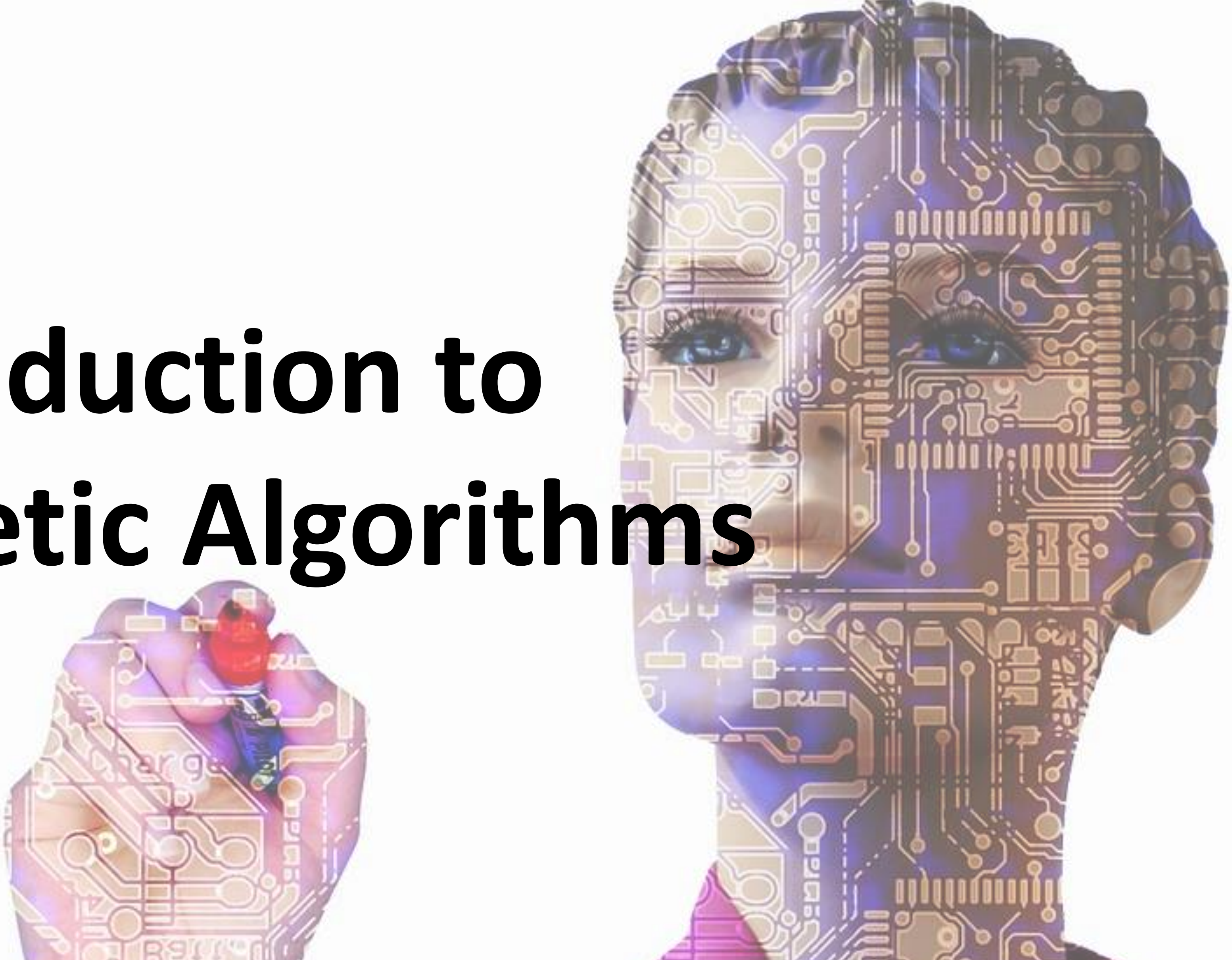
Synthetic Intelligence Lab

Computação Natural@ 2020/2021

- Paulo Novais – pjon@di.uminho.pt
- Filipe Gonçalves – fgoncalves@algoritmi.uminho.pt

- Departamento de Informática
  Escola de Engenharia
  Universidade do Minho
- Grupo ISLab – (Synthetic Intelligence Lab)
- Centro ALGORITMI
  Universidade do Minho

Introduction to Genetic Algorithms



- Training a Machine Learning Model
  - Data:

x1	x2	x3	x4	x5	x6	y
4	-2	7	5	11	1	44.1

$$Y= w1.x1 + w2.x2 + w3.x3 + w4.x4 + w5.x5 + w6.x6$$

  - Goal is to find the set of parameters (w1,w2,w3,w4,w5,w6) that maps the following input to its output:

$$Y' = 4.w1 + (-2).w2 + 7.w3 + 5.w4 + 11.w5 + 1.w6$$

- Solution 1:

w1	w2	w3	w4	w5	w6
2,4	0,7	8	-2	5	1,1

  - Y' = 4.w1 – 2.w2 + 7.w3 + 5.w4 + 11.w5 + w6 ⇔ Y' = 110,3

- Absolute Error:

  Error = | Y – Y' |

  Error = | 44,1 – 110,3 |

  Error = 66,2

- Solution 2:

w1	w2	w3	w4	w5	w6
-0,4	2,7	5	-1	7	0,1

  - Y’ = -0,4.w1 + 2,7.w2 + 5.w3 - 1.w4 + 7.w5 + 0,1.w6 ⇔ Y’ = 100,1

- Absolute Error:

  Error = | Y – Y’ |

  Error = | 44,1 – 100,1 |

  Error = 56

- Solution 3:

w1	w2	w3	w4	w5	w6
-1	2	2	-3	2	0,9

  - Y' = -1.w1 + 2.w2 + 2.w3 - 3.w4 + 2.w5 + 0,9.w6 ⇔ Y' = 13,9

- Absolute Error:

  Error = | Y – Y' |

  Error = | 44,1 – 13,9 |

  Error = 30,2

Difficult to find the best solution manually!

Use optimization Techniques such as **Genetic Algorithms (GA)**

- Genetic Algorithm (GA)
  - Based on Natural Evolution of organisms
  - A brief biological background is helpful to better understand GA

Organism → Cells → Chromosomes → Genes

Eye Color | Hair Color | Freckles | .....

AI Implementation

GA → Individuals/Solutions → Chromosome → Genes

- What are Genes?

  $Y = w1.x1 + w2.x2 + w3.x3 + w4.x4 + w5.x5 + w6.x6$

  - Gene is anything that is able to enhance the results when changed
  - By exploring the following model, the 6 weights are able to enhance the results
    - Thus each weight will represent a gene in GA

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
w1	w2	w3	w4	w5	w6

- Initial Population of Solutions (Generation 0)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
2,4	0,7	8	-2	5	1,1
-0,4	2,7	5	-1	7	$\boxed{0,1}$
-1	2	2	-3	2	0,9
4	7	12	6,1	1,4	-4
3,1	4	0	2,4	4,8	0
-2	3	-7	6	3	3

Chromosome

Gene

- Initial Population of Solutions (Generation 0)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
2,4	0,7	8	-2	5	1,1
-0,4	2,7	5	-1	7	0,1
-1	2	2	-3	2	0,9
4	7	12	6,1	1,4	-4
3,1	4	0	2,4	4,8	0
-2	3	-7	6	3	3

Survival of the Fittest

↓

Fitness Function

↓

Fitness Value

↓

Higher Value => Better Solution

ISLab
Synthetic Intelligence Lab

- Initial Population of Solutions (Generation 0)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y’	F(C)
2,4	0,7	8	-2	5	1,1	110,3	
-0,4	2,7	5	-1	7	0,1		
-1	2	2	-3	2	0,9		
4	7	12	6,1	1,4	-4		
3,1	4	0	2,4	4,8	0		
-2	3	-7	6	3	3		

Y’ = 4.w1 – 2.w2 + 7.w3 + 5.w4 + 11.w5 + w6

Y’ = 4 x 2,4 – 2 x 0,7 + 7 x 8 + 5 x (-2) + 11 x 5 + 1 x 1,1

Y’ = 110,3

- Initial Population of Solutions (Generation 0)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y'	F(C)
2,4	0,7	8	-2	5	1,1	110,3	0,015
-0,4	2,7	5	-1	7	0,1		
-1	2	2	-3	2	0,9		
4	7	12	6,1	1,4	-4		
3,1	4	0	2,4	4,8	0		
-2	3	-7	6	3	3		

$$F(c) = \frac{1}{error} = \frac{1}{|44.1 - 110.3|} = \frac{1}{66.2} = 0.015$$

- Initial Population of Solutions (Generation 0)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y’	F(C)
2,4	0,7	8	-2	5	1,1	110,3	0,015
-0,4	2,7	5	-1	7	0,1	100,1	0,018
-1	2	2	-3	2	0,9	13,9	0,033
4	7	12	6,1	1,4	-4	127,9	0,012
3,1	4	0	2,4	4,8	0	69,2	0,0398
-2	3	-7	6	3	3	3	0,024

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y’	F(C)
2,4	0,7	8	-2	5	1,1	110,3	0,015
-0,4	2,7	5	-1	7	0,1	100,1	0,018
-1	2	2	-3	2	0,9	13,9	0,033
4	7	12	6,1	1,4	-4	127,9	0,012
3,1	4	0	2,4	4,8	0	69,2	0,0398
-2	3	-7	6	3	3	3	0,024

Select best individuals as parents for mating to generate new individuals.

ISLab
Synthetic Intelligence Lab

Computação Natural@ 2020/2021

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y’	F(C)
2,4	0,7	8	-2	5	1,1	110,3	0,015
-0,4	2,7	5	-1	7	0,1	100,1	0,018
-1	2	2	-3	2	0,9	13,9	0,033
4	7	12	6,1	1,4	-4	127,9	0,012
3,1	4	0	2,4	4,8	0	69,2	0,0398
-2	3	-7	6	3	3	3	0,024

Add top 3 individuals to the mating pool for producing the next generation of solutions

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-1	2	2	-3	2	0,9
3,1	4	0	2,4	4,8	0
-2	3	-7	6	3	3

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-1	2	2	-3	2	0,9
3,1	4	0	2,4	4,8	0

- Mating Pool

Crossover

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-1	2	2	-3	2	0,9
3,1	4	0	2,4	4,8	0

Offspring

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-1	2	2	2,4	4,8	0

= 4,8 / 2 = 2,4

Mutation

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-1	2	2	2,4	2,4	0

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	2,4	4,8	0
-2	3	-7	6	3	3

Synthetic Intelligence Lab

Computação Natural@ 2020/2021

- Mating Pool

Crossover

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	2,4	4,8	0
-2	[illegible]	-7	6	3	3

Offspring

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	6	3	3

= 3 / 2 = 1,5

Mutation

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	6	1,5	3

ISLab
Synthetic Intelligence Lab

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-2	3	-7	6	3	3
-1	2	2	-3	2	0,9

- Mating Pool

Crossover

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-2	3	-7	6	3	3
-[illegible]	[illegible]	[illegible]	-3	2	0,9

Offspring

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-2	3	-7	-3	2	0,9

= 2 / 2 = 1

Mutation

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
-2	3	-7	-3	1	0,9

- New Population (Generation 1)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	
-1	2	2	-3	2	0,9	Old Individuals
3,1	4	0	2,4	4,8	0	
-2	3	-7	6	3	3	
-1	2	2	2,4	2,4	0	New Individuals
3,1	4	0	6	1,5	3	
-2	3	-7	-3	1	0,9	

Since there is no garantee that the new individuals will be better than the previous individuals, keeping old individuals saves the results from getting worse

- New Population (Generation 1)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	
-1	2	2	-3	2	0,9	Old Individuals
3,1	4	0	2,4	4,8	0	
-2	3	-7	6	3	3	
-1	2	2	2,4	2,4	0	New Individuals
3,1	4	0	6	1,5	3	
-2	3	-7	-3	1	0,9	

Population → Fitness Values → Mating Pool → Crossover → Mutation

- New Population (Generation 1)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y’	F(C)
-1	2	2	-3	2	0,9	13,9	0,033
3,1	4	0	2,4	4,8	0	69,2	0,04
-2	3	-7	6	3	3	3	0,024
-1	2	2	2,4	2,4	0	44,4	3,333
3,1	4	0	6	1,5	3	53,9	0,102
-2	3	-7	-3	1	0,9	-66,1	0,009

Note: New individuals presented worse results than the old individuals.

- Mating Pool

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	2,4	4,8	0
-1	2	2	2,4	2,4	0
3,1	4	0	6	1,5	3

- Repeat the process to generate new Population (Generation 2)

- New Population (Generation 2)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
3,1	4	0	2,4	4,8	0
-1	2	2	2,4	2,4	0
3,1	4	0	6	1,5	3
3,1	4	0	2,4	1,2	0
-1	2	2	6	0,75	3
3,1	4	0	2,4	2,4	0

Old Individuals

New Individuals

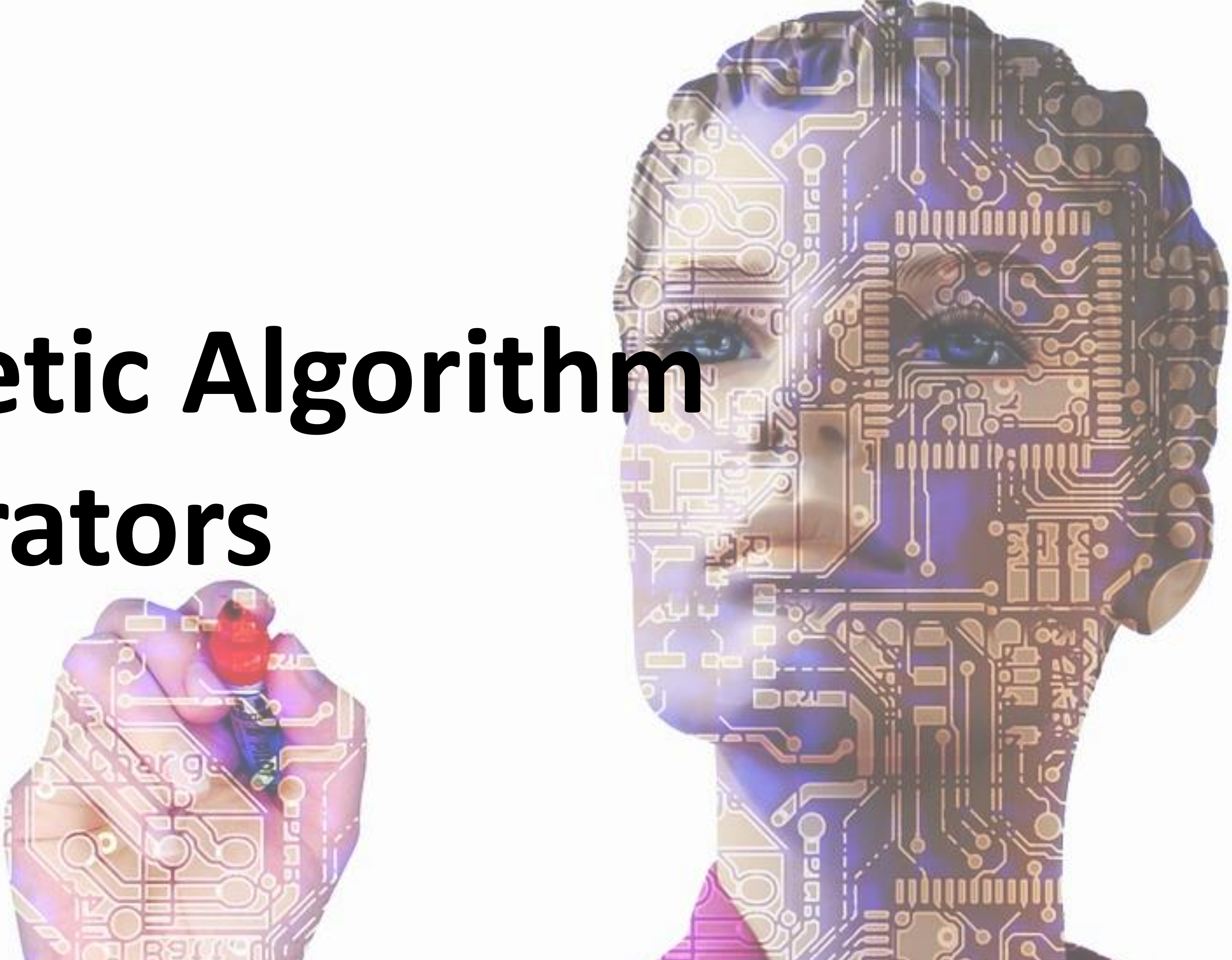
Population → Fitness Values → Mating Pool → Crossover → Mutation

- New Population (Generation 2)

Gene 0	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Y'	F(C)
3,1	4	0	2,4	4,8	0	69,2	0,04
-1	2	2	2,4	2,4	0	44,4	3,333
3,1	4	0	6	1,5	3	53,9	0,102
3,1	4	0	2,4	1,2	0	29,6	0,069
-1	2	2	6	0,75	3	47,25	0,318
3,1	4	0	2,4	2,4	0	42,8	0,77

- Continue the process!

Genetic Algorithm Operators



- **Population Initialization**

There are two primary methods to initialize a population in a GA:

- **Random Initialization:** Populate the initial population with completely random solutions. Random solutions provide diversity of solutions which drive the population to optimal results;
- **Heuristic initialization**: Populate the initial population using a known heuristic for the problem. The entire population should not be initialized using a heuristic, as it can result in the population having similar solutions and very little diversity.

- **Population Models**

There are two population models widely in use:

- **Steady State:** generate one or two off-springs in each iteration and they replace one or two individuals from the population (also known as Incremental GA);
- **Generational:** generate 'n' off-springs, where n is the population size, and the entire population is replaced by the new one at the end of the iteration.

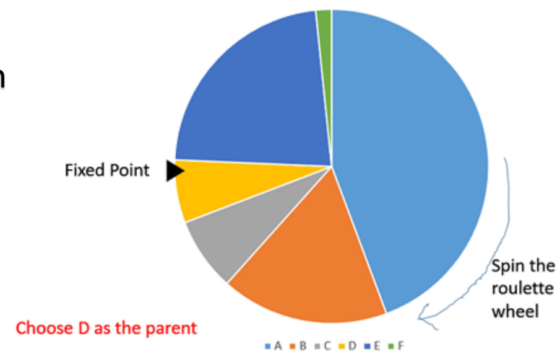
- **Fitness Function:** Defines a function which takes a candidate solution to the problem as input and produces as output how “fit”/“good” the solution is with respect to the problem in consideration.

- **Take into account:**
  - Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast;
  - In most cases the fitness function and the objective function are the same as the objective is to either maximize or minimize the given objective function;
  - For more complex problems with multiple objectives and constraints, an Algorithm Designer might choose to have a different fitness function.

- **Parent Selection:** Process of selecting parents which mate and recombine to create off-springs for the next generation. Maintaining good diversity in the population is extremely crucial for the success of a GA.

- **Parent Selection – Operators:**
  - **Fitness Proportionate Selection:**
    - **Roulette Wheel Selection**
    - **Stochastic Universal Sampling (SUS)**
  - **Tournament Selection**
  - **Rank Selection**
  - **Fitness Selection**
  - **Random Selection**

- **Parent Selection – Operators:**
  - **Fitness Proportionate Selection:**
- **Roulette Wheel Selection:**
  - Consider a circular wheel. The wheel is divided into n pies, where n is the number of individuals in the population;
  - Each individual gets a portion of the circle which is proportional to its fitness value. A fixed point is chosen on the wheel is rotated;
  - The region of the wheel which comes in front of the fixed point is chosen as the parent. For the second parent, the same process is repeated.

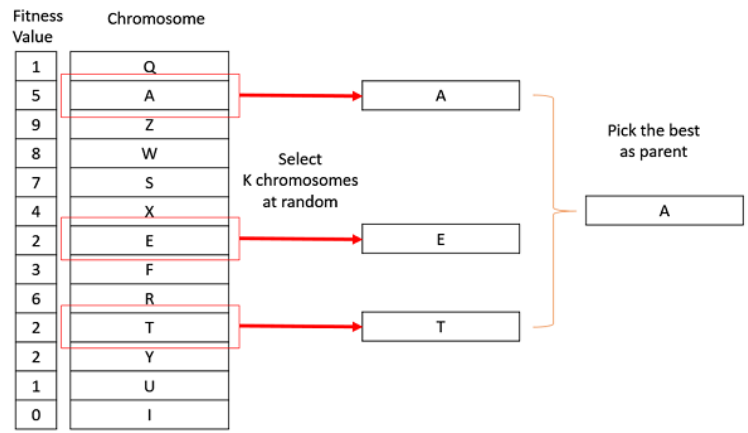


Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

- **Stochastic Universal Sampling (SUS):** Similar to Roulette Wheel Selection, with the difference of presenting multiple fixed points. All parents are chosen in just one spin of the wheel.

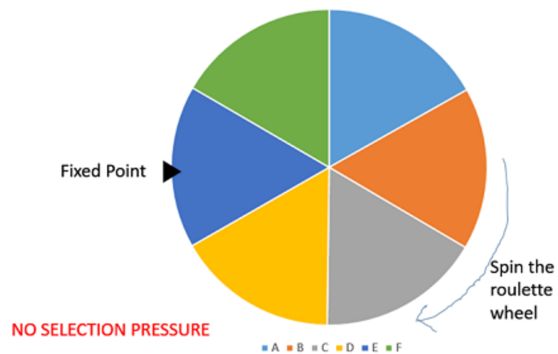
- **Parent Selection – Operators:**

  ○ **Tournament Selection:** n K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent.



  ○ **Rank Selection:**

  - Mostly used when the individuals in the population have very close fitness values;
  - Every individual in the population is ranked according to their fitness, where the higher ranked individuals are preferred more than the lower ranked ones.



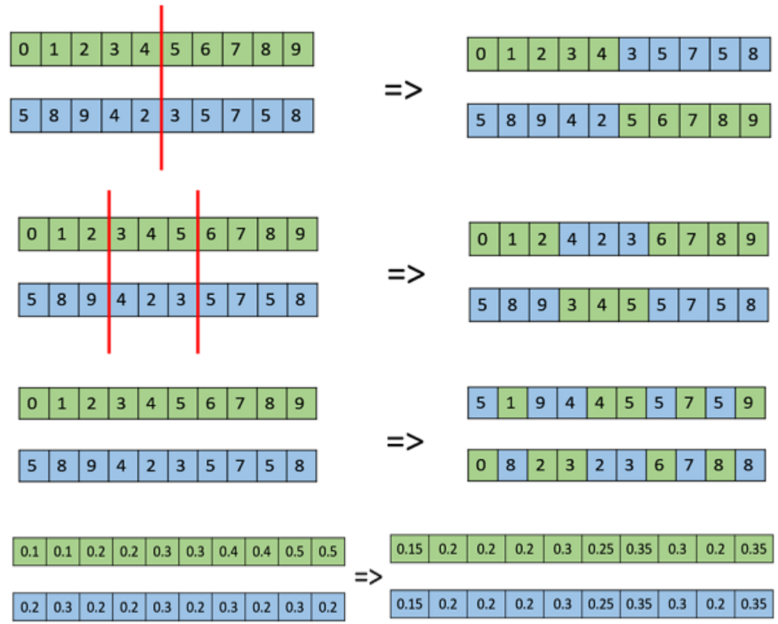
Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

- **Parent Selection – Operators:**

- **Fitness Selection:** select parents from the existing population with the best fitness values.

- **Random Selection:** randomly select parents from the existing population. There is no selection pressure towards fitter individuals and therefore this strategy is usually avoided.

- **Crossover:** creation of the new off-springs by combining the genes of the parents' chromosomes
- **Crossover - Operators:**
  - **One Point Crossover:** a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.
  - **Multi Point Crossover:** a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.
  - **Uniform Crossover:** each gene is threated separately by "flipping a coin" to decide whether it'll be included in the off-spring.
  - **Whole Arithmetic Recombination:** works by taking the weighted average of the two parents.



- **Mutation:** random tweak in the chromosome to get a new solution, used to maintain and introduce diversity in the genetic population.
- **Mutation – Operators:**
  - **Bit Flip Mutation:** select one or more random bits and flip them. This is used for binary encoded GAs.

    0 0 1 **1** 0 1 0 0 1 0 => 0 0 1 **0** 0 1 0 0 1 0

  - **Random Resetting:** a random value from the set of permissible values is assigned to a randomly chosen gene.

    1 **2** 3 4 5 **6** 7 8 9 0 => 1 **6** 3 4 5 **2** 7 8 9 0

  - **Swap Mutation:** select two positions on the chromosome at random and interchange the values.

    1 **2** 3 4 5 **6** 7 8 9 0 => 1 **6** 3 4 5 **2** 7 8 9 0

  - **Scramble Mutation:** select a subset of genes and their values are scrambled or shuffled randomly.

    0 1 **2 3 4 5 6** 7 8 9 => 0 1 **3 6 4 2 5** 7 8 9

  - **Inversion Mutation:** select a subset of genes and invert the entire string in the subset.

    0 1 **2 3 4 5 6** 7 8 9 => 0 1 **6 5 4 3 2** 7 8 9

- **Survivor Selection:** determines which individuals are to be kicked out and which are to be kept in the next generation.
- **Survivor Selection – Operators:**
  - **Age Based Selection:** based on the premise that each individual is allowed in the population for a finite generation where it is allowed to reproduce, after that, it is kicked out of the population no matter how good its fitness is.
  - **Fitness Based Selection:** the children tend to replace the least fit individuals in the population. The selection of the least fit individuals may be done through the application of a selection policies.

- **Termination Condition:** trigger to stop the GA from running when new solutions don't provide considerable improvements.

- Usually, we keep one of the following termination conditions:
  - When there has been no improvement in the population for X iterations.
  - When the algorithms reaches an absolute number of generations.
  - When the objective function value has reached a certain pre-defined value.

- **Genetic Algorithms:**

○ **Advantages:**

- Does not require any derivative information.
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Useful when the search space is very large and there are a large number of parameters involved.

○ **Limitations:**

- GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

- **Genetic Algorithms applied on:**

  - Optimization
  - Economics
  - Neural Networks
  - Parallelization
  - Image Processing
  - Vehicle routing problems
  - Scheduling applications
  - Machine Learning
  - Robot Trajectory Generation
  - Parametric Design of Aircraft
  - DNA Analysis
  - Multimodal Optimization
  - Traveling salesman problem and its applications

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática
Computação Natural
2020/2021

Filipe Gonçalves, Paulo Novais