LECTURER: Nghia Duong-Trung

MACHINE LEARNING

MACHINE LEARNING
TOPIC OUTLINE

UNIT 6

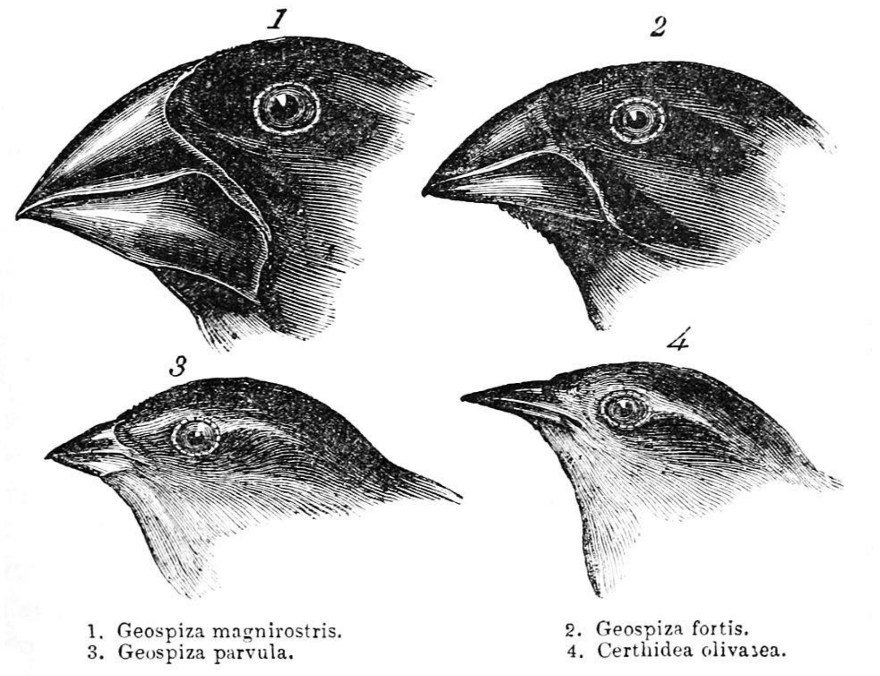
GENETIC ALGORITHMS

STUDY GOALS

- Know the definitions and terms used for evolutionary algorithms including Genetic Algorithms (GA)
- Comprehend the important concepts of GA
- Understand the main phases of GA
- Apply GA for the Knapsack problem
- Implement GA in Python

Evolution approaches:

- **Genetic** algorithms: simulates the **reproduction of individuals** in a population in order to solve a specific optimization problem.
- **Swarm** algorithms: Simulate the ***movement*** of a group of animals toward a specific target.
- **Ant Colony** algorithms: simulate the ***communication*** between insects to find the shortest path between their nest and the food source.

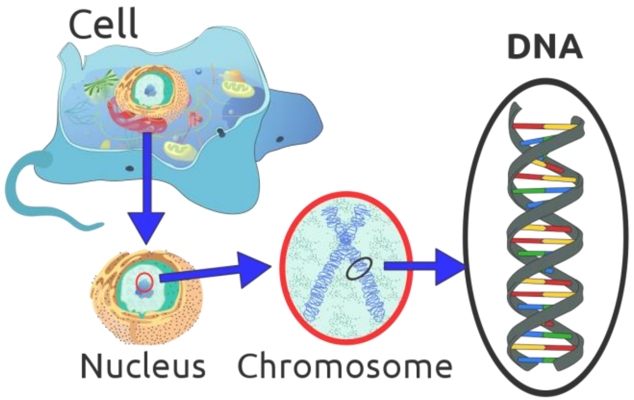


Example: Evolutionary biology

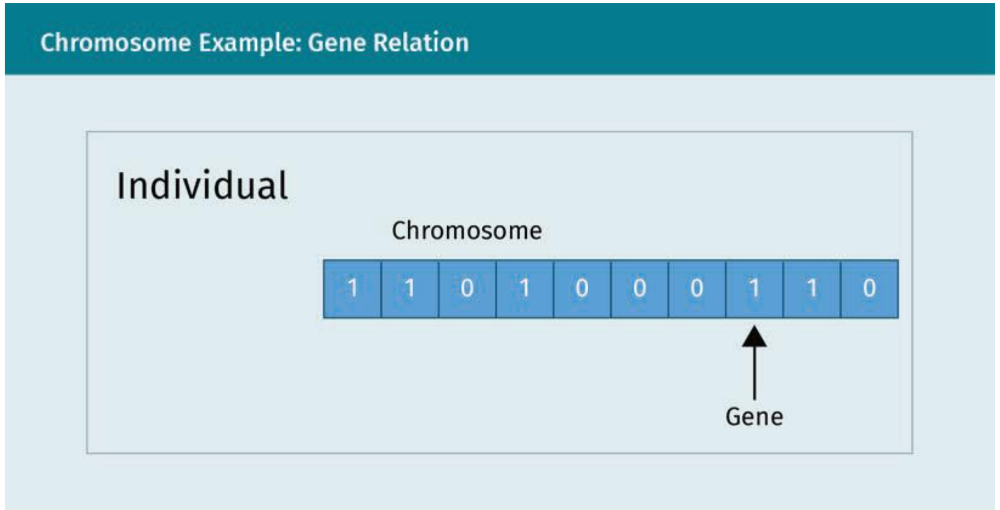
Source of the text: Zöller, 2022.
Source of the image: File: Darwin's_finches_by_Gould.jpg (2023).

Genetic algorithms (GA):

- Biological genetics: An individual in a **population** is made-up of cells encoded by **genes** (**DNA**) in **chromosomes.**
- GA in **Machine Learning**:
  - are inspired by the biological **genetic evolution** of living organisms
  - to find the **optimum** or near-optimum solutions
  - By **iteratively** exploring possible solutions
  - DNA and chromosome are encoded as **bits**



Organization of DNA in a eukaryotic cell



Chromosome representation in Genetic Algorithms

Source of the text: Sammut, 2011; Zöller, 2022.
Source of the image: File: Eukaryote_DNA-en.svg (2023); Zöller (2022, p.109).

Basic terms:

- Individual: any possible solution
- Population: group of all individuals within an investigation problem
- Fitness: target function that we are optimizing (each individual has a fitness)

The output of the genetic algorithm is a quantity

Basic process:

- It starts from a population of randomly generated individuals and happens in generations.
- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected (based on their fitness), and modified to form a new population.
- Next-generation competes with each other, the process goes on until the perfect program is evolved.
- The new population is used in the next iteration of the algorithm.
- The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

Applications of GA

- Problem-solving engine
- Curve fitting
- Data modeling
- Automatic programming
- Symbolic regression
- Feature selection
- Classification

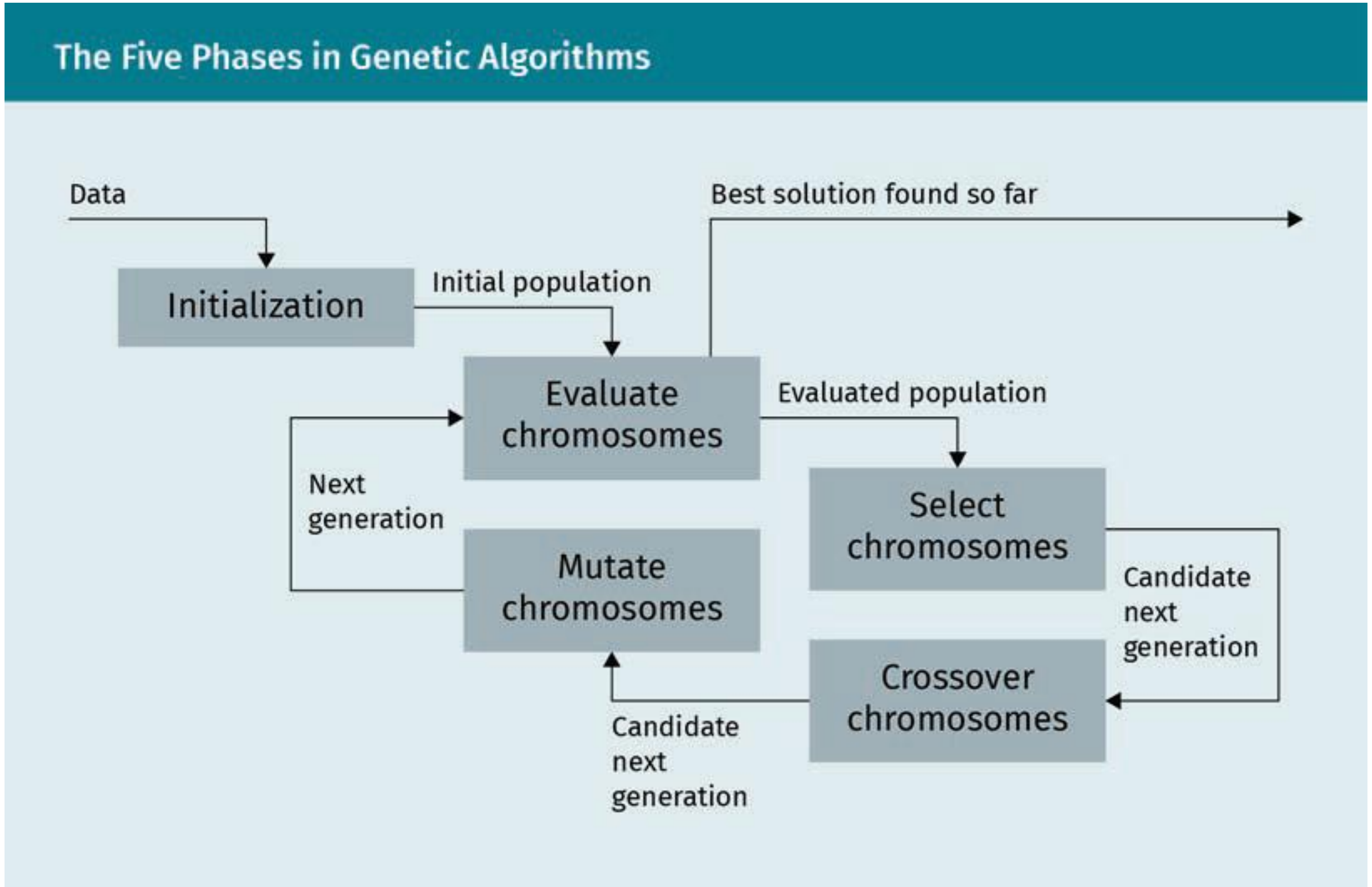
Methods of GA

- SIMPLE GA
- REAL-CODED GA
- HIERARCHICAL FAIR COMPETITION GA
- MEMETIC GA

Source of the text: Chaiyaratana & Zalzala, 1997; Genetic programming , 2023; Sivanandam & Deepa, 2008; Zöller, 2022.

GA Phases

- Initialization
- Evaluation
- Selection
- Crossover
- Mutation

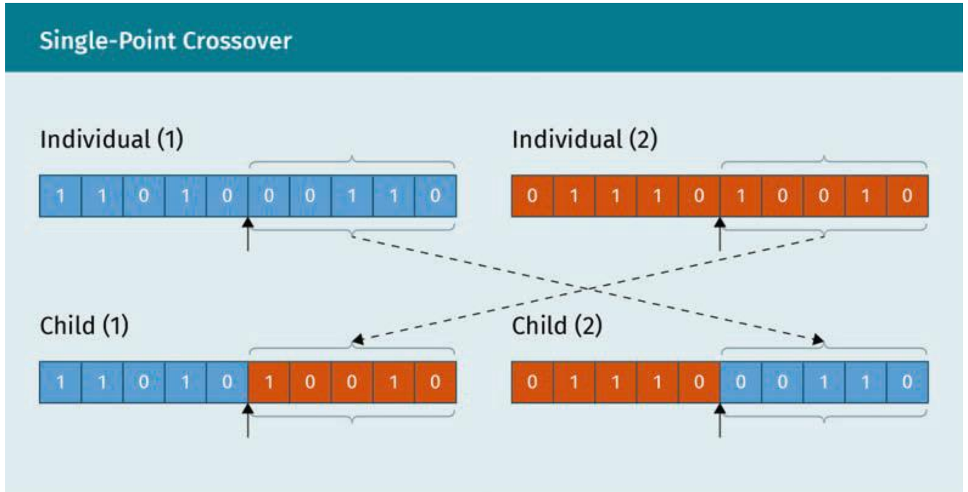


The five phases in Genetic Algorithms

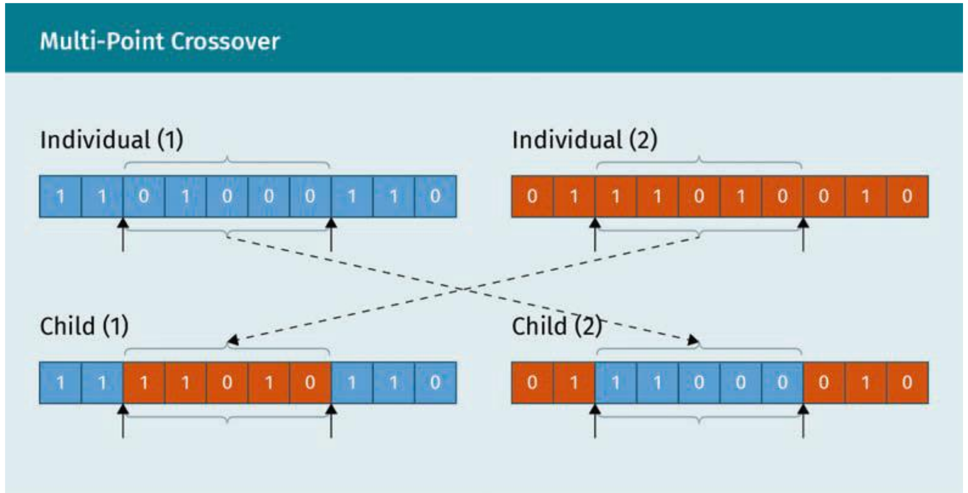
Source of the text: Sammut, 2011; Zöller, 2022.
Source of the image: Zöller (2022, p. 133).

Crossover phase:

- Function: **Mixing** of the genes from parent‘s pair of chromosomes
- Effect: the child **shares** the characteristics of both parents
- Crossover point: **random** point in the chromosome, where the gene thread are cut and exchanged
- Types:
  - **Single**-point crossover: one locus is selected
  - **Multiple**-point crossover: a set of crossover points are selected



Single-point crossover in genetic algorithms

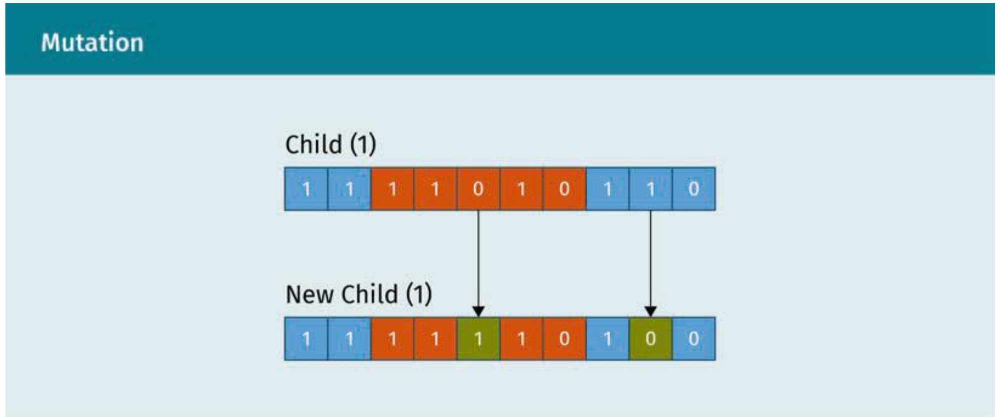


Multiple-point crossover in genetic algorithms

Source of the text: Sivanandam & Deepa, 2008; Zöller, 2022.
Source of the image: Zöller (2022, p. 134).

Mutation phase

- Function: **flipping** of **random** genes at arbitrary location(s) in an individual chromosome
- Effect: **prevent** falling into **local optimum** solutions too quickly
- Types of changes:
  - **Deletion**
  - **Insertion**
  - **Re-arrangement**



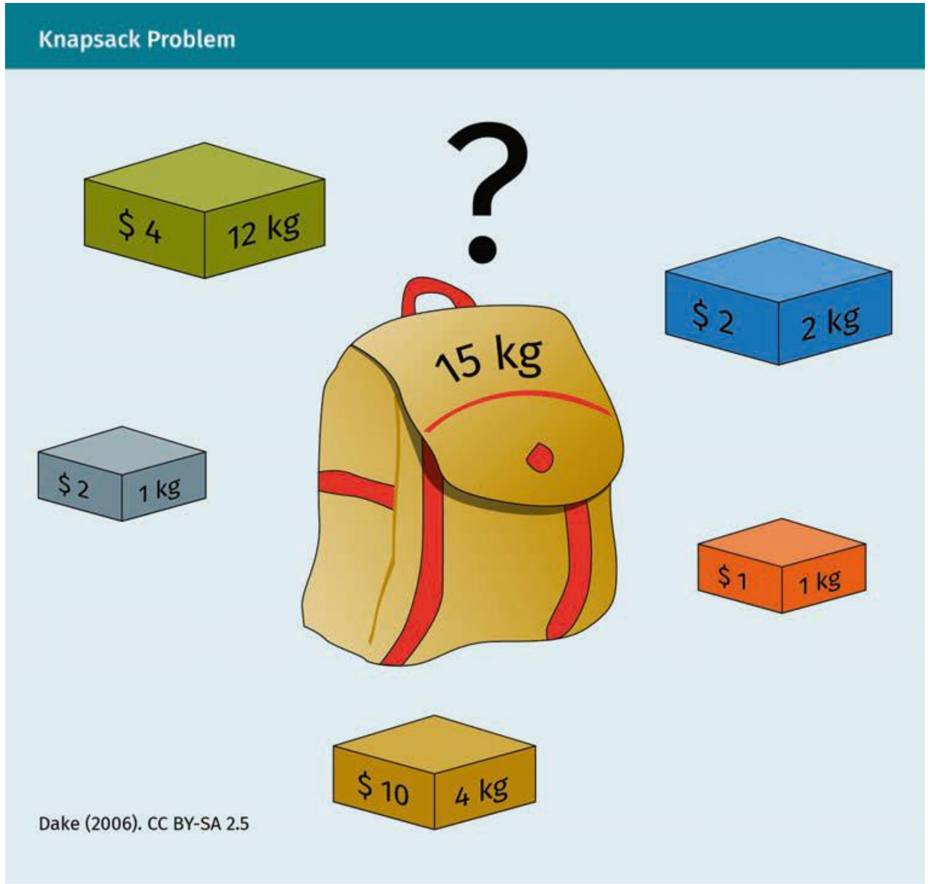
Mutation in genetic algorithms

Source of the text: Sivanandam & Deepa, 2008; Zöller, 2022.
Source of the image: Zöller (2022, p. 135).

Knapsack Problem

- Given:
  - A set of items, each has weight and value
  - A knapsack of a max capacity
- Problem:

  Select a subset of items so that
  - The total value is as large as possible
  - Total weight ≤ max capacity



An example of genetic algorithms: Knapsack problem

Source of the text: Zöller, 2022.
Source of image: Zöller (2022, p. 136).

THE KNAPSACK PROBLEM

W: 7kg	W: 2kg	W: 1kg	W: 9kg
V: 5€	V: 4€	V: 7€	V: 2€

15kg

0	1	1	0	W: 3kg	V: 11€
1	1	0	1	18kg	0€

Source of image: https://uxwing.com/backpack-icon/

W: 7kg	W: 2kg	W: 1kg	W: 9kg
V: 5€	V: 4€	V: 7€	V: 2€

An initial population is 8 solutions.

0	1	0	1		V: 6€
0	0	0	1		2
1	1	1	1		0
0	1	1	1		13
0	0	1	1		9
0	0	0	0		0
0	1	1	0		11
0	0	0	1		2

W: 7kg	W: 2kg	W: 1kg	W: 9kg
V: 5€	V: 4€	V: 7€	V: 2€

Randomly select 2 solutions and they compete with each other.
The winner becomes a parent.

0	1	0	1		**V: 6€**
0	0	0	1		2
1	1	1	1		0
0	1	1	1		13
0	0	1	1		9
0	0	0	0		0
0	1	1	0		11
0	0	0	1		2

W: 7kg	W: 2kg	W: 1kg	W: 9kg
V: 5€	V: 4€	V: 7€	V: 2€

Randomly select 2 solutions and they compete with each other.
The winner becomes a parent.

0	1	0	1		V: 6€
0	0	0	1		2
1	1	1	1		0
0	1	1	1		13
0	0	1	1		**9**
0	0	0	0		0
0	1	1	0		11
0	0	0	1		2

Crossover

0	1	0	1
0	0	1	1

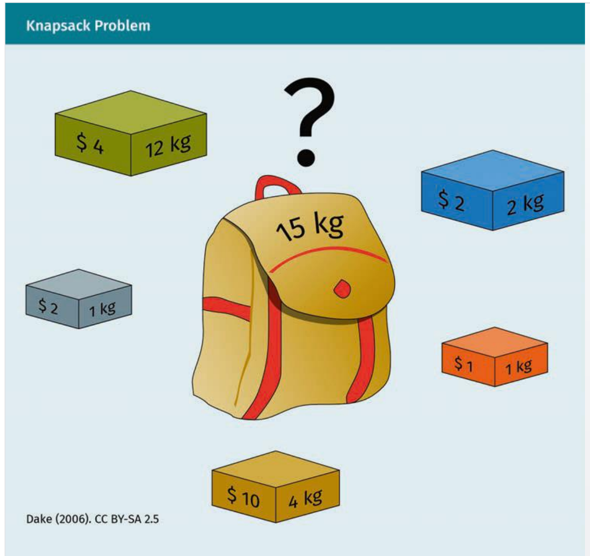
Mutation

0	1	1	1
0	0	0	1

0	1	1	0
0	0	0	1

Repeat to have the same amount as the initial population

6.3 GENETIC ALGORITHM IN PYTHON



Definitions:

Kcap – Max capacity

POP_SIZE – Number of solutions

GEN_MAX – Number of generations

NUM_ITEMS – Number of items

initialization() – Creating an initial population

fitness() – Measuring the fitness of an individual

evolution() – Selection, crossover, mutation

```
# Main function of genetic algorithm in python
def main():
   import random
   KCap = 30; POP_SIZE = 30; GEN_MAX = 50; NUM_ITEMS = 15
   ITEMS = [(random.randint(0,20),random.randint(0,20))
              for x in range (0, NUM_ITEMS)]
   generation = 1 #Generation counter
   population = initialization(POP_SIZE, NUM_ITEMS)
   for g in range(0,GEN_MAX):
       totalFitness = 0
       for i in population:
           totalFitness += fitness(i, ITEMS, KCap)
       print(totalFitness)
       population = evolution(population)
       generation += 1
   population = sorted(population,
                       key=lambda ind: fitness(ind, ITEMS, KCap),
                       reverse=True)
   print(population[0])
```

Source of the text: Gad, 2018; Zöller, 2022.
Source of the image: Zöller (2022, p. 136).

REVIEW STUDY GOALS

- Know the definitions and terms used for evolutionary algorithms including Genetic Algorithms (GA)
- Comprehend the important concepts of GA
- Understand the main phases of GA
- Apply GA for the Knapsack problem
- Implement GA in Python

SESSION 6

GENETIC ALGORITHMS

Implement the main() function of GA for solving the Knapsack Problem with the following inputs:

- Max carry weight: M = 30
- Number of available items: n = 7,
- Each item has random weight ($0 \leq w \leq 15$) and value ($0 \leq v \leq 10$)
- Number of randomly generated chromosomes: m = 10
- Number of generations: G = 100

Please present your results.

The results will be discussed in plenary.

1. Genetic, swarm and ant colony algorithms are examples of _____ algorithms.

   a) population-based
   b) metaheuristic
   c) optimization
   d) all of the options are true.

2. Genetic algorithms consist of five main phases in this order:

a) (1) Initialization, (2) crossover, (3) evaluation, (4) selection of the fittest, and (5) mutation of the genetic structure.
b) (1) Initialization, (2) evaluation, (3) selection of the fittest, (4) crossover, and (5) mutation of the genetic structure.
c) (1) Initialization, (2) mutation of the genetic structure, (3) crossover, (4) evaluation, and (5) selection of the fittest.
d) (1) Initialization, (2) selection of the fittest, (3) evaluation, (3) crossover, and (4) mutation of the genetic structure.

3. The ______ can help to escape local optimum solution in the population.

a) mutation phase
b) crossover phase
c) evaluation phase
d) selection phase

4. In genetic algorithms, each chromosome represents ...

a) ... the optimal solution for the domain problem.
b) ... the fitness of a specific solution for the domain problem.
c) ... a part of the solution for the domain problem.
d) ... a possible solution for the domain problem.

5. Genetic algorithms________ the finding of the optimal solution for a given problem.

a) guarantee
b) are not established for
c) do not guarantee
d) none of these

LIST OF SOURCES

Text:

Chaiyaratana, N., & Zalzala, A. M. S. (1997). Recent developments in the evolution strategies of genetic algorithms: Theory and applications. *Research Report. ACSE Research Report 666.*

Gad, A. (2018). Genetic algorithm implementation in Python. https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6

Genetic programming (2023, February 13). In *Wikipedia.* Retrieved, February 22, 2023, from https://en.wikipedia.org/wiki/Genetic_programming

Sammut, C. (2011). Genetic and evolutionary algorithms. In C. Sammut & G. I. *Webb(Eds.), Encyclopedia of machine learning. Springer Science & Business Media.*

Sivanandam S.N., Deepa S.N.(2008).Classification of Genetic Algorithm. Pp 105-129. Springer. https://doi.org/10.1007/978-3-540-73190-0_5

Zöller, T. (2022). Course Book – Machine Learning. *IU International University of Applied Science.*

Images:

File: Darwin's_finches_by_Gould.jpg (February 20, 2023). *In Wikipedia Commons, the free media repository*. Retrieved, February 22, 2023, from https://en.wikipedia.org/wiki/Evolution

File: Eukaryote_DNA-en.svg (January 20, 2023). *In Wikipedia Commons, the free media repository*. Retrieved, February 22, 2023, from https://en.wikipedia.org/wiki/Chromosome

File: Genetic_programming_subtree_crossover.gif. (February 13, 2023). *In Wikipedia Commons, the free media repository*. Retrieved, February 22, 2023, from https://en.wikipedia.org/wiki/Genetic_programming

Zöller (2022, p. 132).

Zöller (2022, p. 133).

Zöller (2022, p. 134).

Zöller (2022, p. 135).

Zöller (2022, p. 136).