5. Exercise Sheet

Exercise 1 Evolutionary Algorithms - Definition and Example

The **knapsack problem** is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. — *Wikipedia, 06.06.2018*

a) Describe and sketch the basic structure of an evolutionary algorithm (EA).

b) How can we solve an instance of the Knapsack Problem (see description below) using an EA. Define suitable representation, fitness function, and genetic operators.

Exercise 2 Genetic Representation and Genetic Operators

In the lecture we described different encodings for the representation of a solution for the 8-Queens problem, namely

- Binary matrix with up to 64 queens
- Position vector with 8 entries
- Binary matrix with exactly 8 queens
- Integer vector, 1 Queen per row
- Permutation, 1 Queen per row and column

Provide a suitable cross-over and mutation operator for each of them. How does your operator (visually) effect the board state?

Exercise 3 Roulette Wheel Selection

a) Given the fitness distribution in the table below, determine the probability that an individual will be chosen using roulette wheel selection

Individual	1	2	3	4	5	6	7
Fitness	60	250	320	140	80	150	20

b) Determine the probability that an individual with fitness p was chosen k times to be put into the mating pool.

c) How does the probability change if the same individual is included multiple times in the current population.

d) How do we need to adapt the fitness calculation if the fitness of individuals should be minimized.

Note: Using Roulette Wheel Selection we assign a probability to each individual and repeatedly sample an individual from the population with replacement.

Exercise 4 Tournament Selection

a) What is the probability of the best individual to get into the mating pool in case we have a population and mating pool size of 10 and a tournament size of 4.

b) How does the probability change if we change the tournament size to 6. How does it change in general?

c) What is the expected number of copies of the best individual in the mating pool?

d) What is the probability of the worst individual to get into the mating pool?

Note: In a single tournament an individual can only participate once, therefore, we choose a sample without replacement of the population to participate in the tournament:

The following task can be solved in pairs of two. Please make sure that your solution includes the name of both group members as a comment at the top of the file

Exercise 5 Programming Exercise - Implementing NEAT for Flappy Bird

After we discussed all the components of an EA in detail want to apply it to solve game related problems. In the following we will have a look on Flappy Bird and how to play it using the NEAT algorithm. Neuroevolution of augmenting topologies (NEAT) is a genetic algorithm (GA) for the generation of evolving artificial neural networks. The input of the network is the representation of the current game state and its output layer determines the agents action. Your task will be to implement the GA algorithm including a fitness measure,

Joining the Google Colab project

- Join the Google Colab project:
  `https://colab.research.google.com/drive/1kP9VniakGdSYdRuAFQdWg4d4IjKH3xjo`

Accessing all files on Github

- Link to files on GitHub; Requirements: Python 3.5 or higher, packages: gym, numpy
  `https://github.com/ADockhorn/NEATCIG`

Included files

- *main.py* includes the experiment setup
- *population.py* provides access points for a basic evolutionary algorithm
- Agents:
  - *randomagent.py* picks a random action in each timestep
  - *naiveagent.py* defines a very basic action selection strategy which tries to stay above the next pipe's bottom edge.
  - *nnagent.py* provides a basic implementation of a neural network and respective crossover and mutate functions

Task

a) Read up on the NEAT algorithm

b) Implement the NEAT algorithm in population.py using the neural network agent provided by nnagent.py

c) Analyze the learning behavior (maximal, minimal and average fitness over time) and compare the result to the given heuristic in naiveagent.py

d) Can you find a better heuristic than the one provided in naiveagent.py?