Carrying out experiments with genetic algorithms

Example using the *Knapsack* problem

1 – Creating the project executable

- Steps:
  - Access the command line and position yourself in the python folder:

  For example: *cd C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts*
  - Install *pyinstaller* using pip:

  pip install pyinstaller
  - Position yourself in the project folder on which you want to create the executable:

  cd folder_path
  - Run the following command to create the executable:

  C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts\pyinstaller --onefile main.py

  Note: The main.exe file will be created in your project's dist folder

  It may be necessary to previously install the numpy and matplotlib libraries using pip: *pip install numpy / pip install matplotlib*

  It may be necessary to incorporate the *matplotlib* library using:

  pyinstaller --hidden-import matplotlib.backends.backend_tkagg --onefile main.py

2 – Creating the folder structure for the experiments

Tests

- *Dataset1*
  - *GeneralTests*
  - *Tournament*
  - *Population*
  - *Recombination*
  - *Mutation*
- *Dataset2*
  - *GeneralTests*
  - *Tournament*
  - *Population*
  - *Recombination*
  - *Mutation*
- *…*

Files to place inside each subfolder:

(the .exe file is placed inside each subfolder, so that each type of experiment is saved in different excel files)

data_set_1.txt
main.exe
config.txt

data_set_2.txt
main.exe
config.txt

…

3 – Performing experiments

Consider the project developed in class for the Knapsack problem (main.exe executable file)

1 – Start by carrying out preliminary analyzes on each dataset, in order to find out the values of the parameters that are best suited to solving the problem. In this analysis, it should be possible to identify the appropriate number of generations (population size and number of generations depend on the complexity of the optimization problem and must be determined experimentally for each dataset)
2 – Perform (automated) experiments with the variation slightly above/below the values found to determine the best combination of parameters. Use between 30 to 50 runs for each combination of parameters
3 – Carry out (automated) experiments that vary the various GA parameters, to produce graphs that show the influence of their values on the fitness average:

- Populaçtion vs generations
- Tournament size
- Recombination methods and their probabilities
- Mutation methods and their probabilities

3.1 – General tests for each dataset

- After preliminary analyzes of the dataset, it is necessary to define the config file that will be used by the application to carry out multiple tests with various combinations of parameters. This file must be placed inside the GeneralTests folder of Dataset1
- In this example (config file shown on the right) 50 runs will be performed for each parameter combination, for Dataset1
- In total this example will allow 16200 experiments to be carried out (324 parameter combinations * 50 runs)
- The results of these experiments are saved in 3 files:
  - *statistic_average_fitness*
  - *statistic_best_per_experiment*
  - *statistic_best_per_experiment_fitness*

- The file that should be used for analysis of the best combination of parameters should be the file statistic_average_fitness

```
Runs: 50

Population_size: 50, 100, 200

Max_generations: 50

# ----------------------------------

Selection: tournament

Tournament_size: 2, 4, 6, 8

# ----------------------------------

Recombination: one_cut, two_cuts, uniform

Recombination_probability: 0.6, 0.7, 0.8

# ----------------------------------

Mutation: binary

Mutation_probability: 0.01, 0.025, 0.03

# ----------------------------------

Probability_of_1s: 0.05

Fitness_type: 0

# ----------------------------------

Problem_file: ./data_set_1.txt

# ----------------------------------

Statistic: BestIndividual
Statistic: BestAverage
```

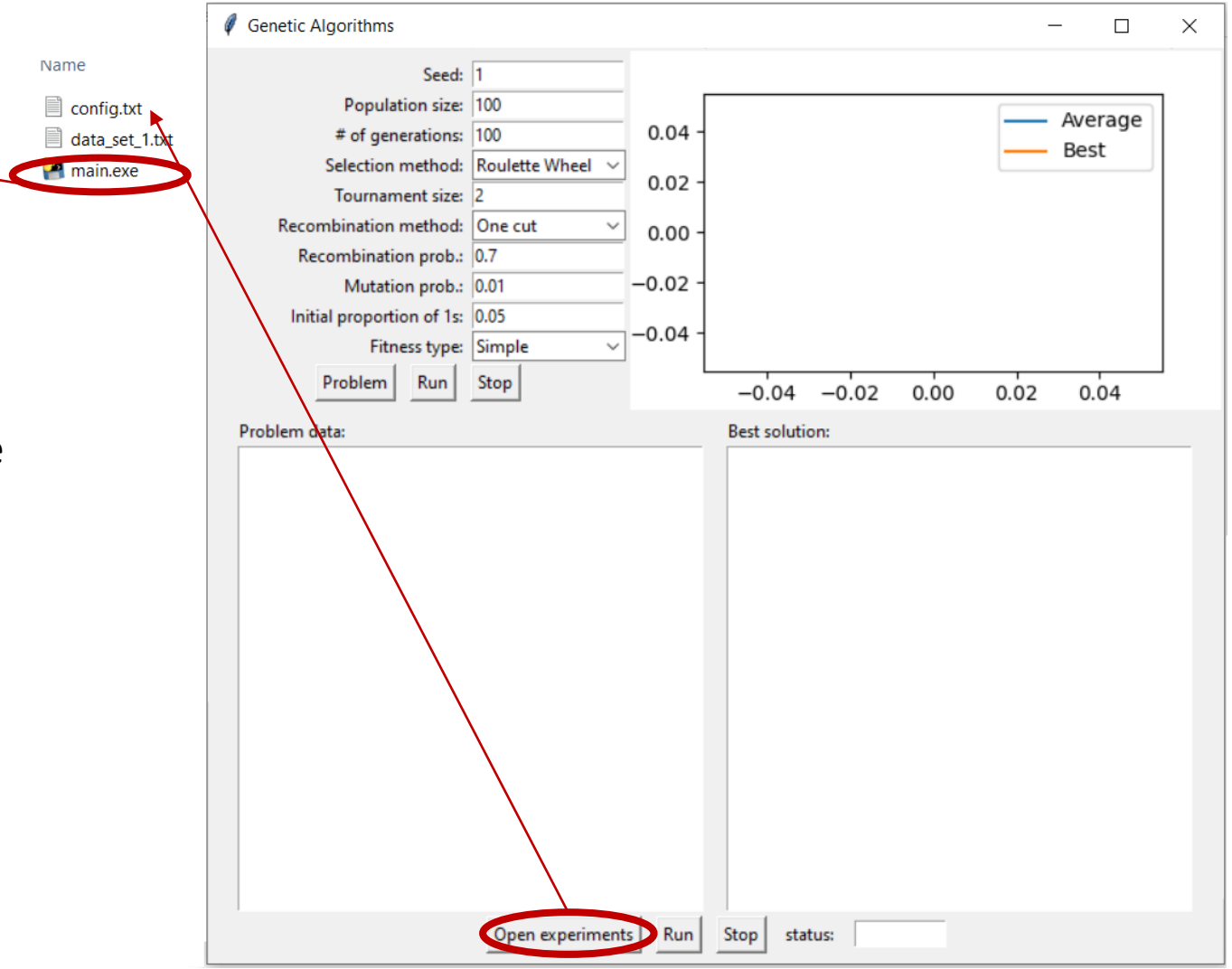
3.1 – General tests for each dataset (cont.)

1 – Run the .exe file

2 – Click on *Open experiments* and select the *config* file

3 – Click on *Run*

- *status: running* will be displayed right after the start of the experiments
- Upon completion of the experiments, the *status: finished* will be displayed
- The files generated by the experiments must not be opened until the program has finished the experiments (if you want to consult their contents, copy the files to a temporary folder)



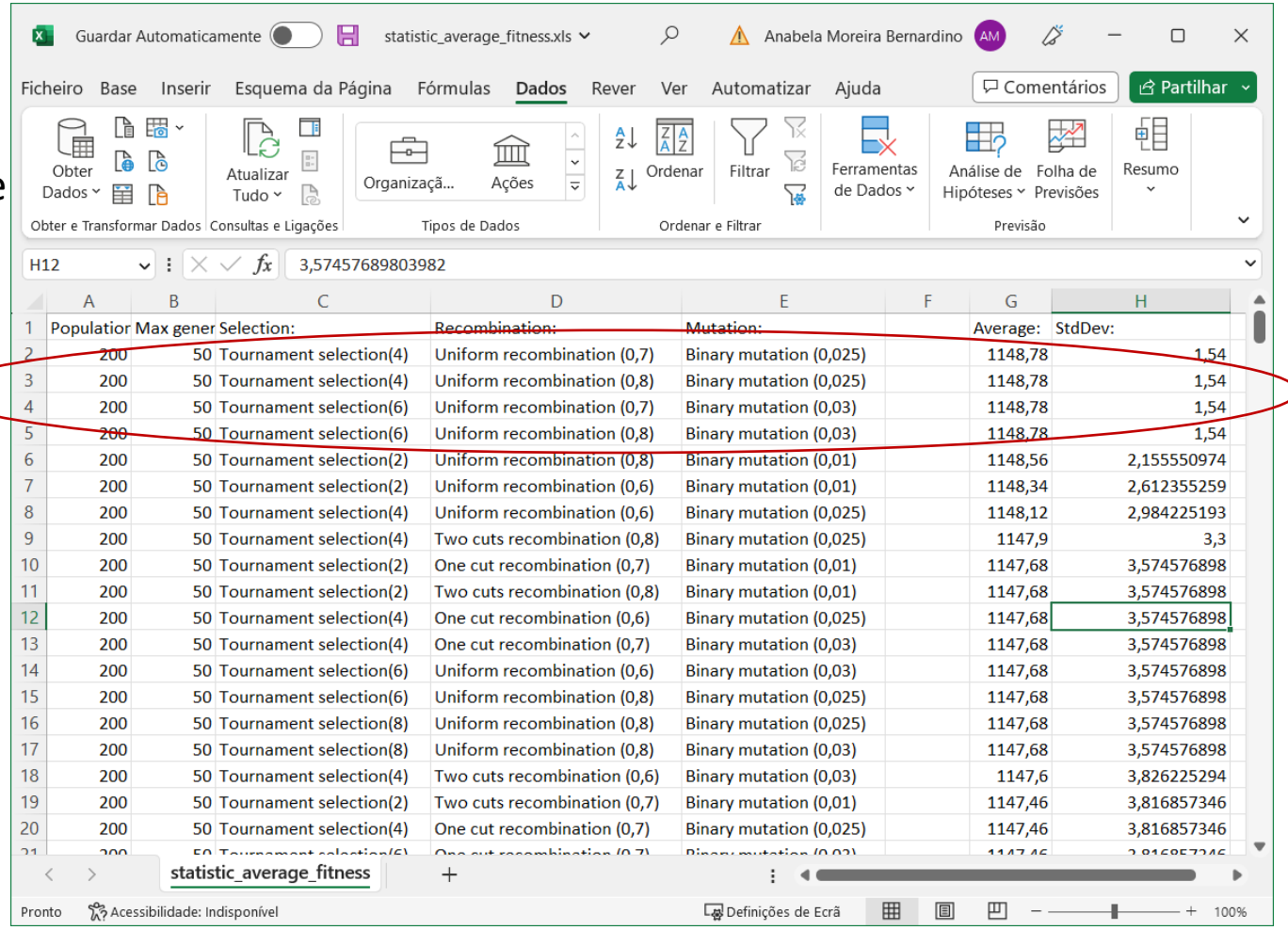
3.1 – General tests for each dataset (cont.)

1 – Open the *statistic_average_fitness* file

2 – Verify that the data in the Average column appears right-aligned. If they are not aligned to the right, it is necessary to change the formatting of the numbers in the Excel options, or simply select the column data and replace '.' with ','

3 – Sort the data in descending order of the Average column (in the Knapsak problem the best combination of parameters is the one with the highest average fitness)

As can be seen, the best result obtained was for the combination of parameters *population 200, tournament 4/6, uniform recombination 0.7/0.8 and binary mutation 0.025/0.03*



	A	B	C	D	E	F	G	H
1	Populatior	Max gener	Selection:	Recombination:	Mutation:		Average:	StdDev:
2	200	50	Tournament selection(4)	Uniform recombination (0,7)	Binary mutation (0,025)		1148,78	1,54
3	200	50	Tournament selection(4)	Uniform recombination (0,8)	Binary mutation (0,025)		1148,78	1,54
4	200	50	Tournament selection(6)	Uniform recombination (0,7)	Binary mutation (0,03)		1148,78	1,54
5	200	50	Tournament selection(6)	Uniform recombination (0,8)	Binary mutation (0,03)		1148,78	1,54
6	200	50	Tournament selection(2)	Uniform recombination (0,8)	Binary mutation (0,01)		1148,56	2,155550974
7	200	50	Tournament selection(2)	Uniform recombination (0,6)	Binary mutation (0,01)		1148,34	2,612355259
8	200	50	Tournament selection(4)	Uniform recombination (0,6)	Binary mutation (0,025)		1148,12	2,984225193
9	200	50	Tournament selection(4)	Two cuts recombination (0,8)	Binary mutation (0,025)		1147,9	3,3
10	200	50	Tournament selection(2)	One cut recombination (0,7)	Binary mutation (0,01)		1147,68	3,574576898
11	200	50	Tournament selection(2)	Two cuts recombination (0,8)	Binary mutation (0,01)		1147,68	3,574576898
12	200	50	Tournament selection(4)	One cut recombination (0,6)	Binary mutation (0,025)		1147,68	3,574576898
13	200	50	Tournament selection(4)	One cut recombination (0,7)	Binary mutation (0,03)		1147,68	3,574576898
14	200	50	Tournament selection(6)	Uniform recombination (0,6)	Binary mutation (0,03)		1147,68	3,574576898
15	200	50	Tournament selection(6)	Uniform recombination (0,8)	Binary mutation (0,025)		1147,68	3,574576898
16	200	50	Tournament selection(8)	Uniform recombination (0,8)	Binary mutation (0,025)		1147,68	3,574576898
17	200	50	Tournament selection(8)	Uniform recombination (0,8)	Binary mutation (0,03)		1147,68	3,574576898
18	200	50	Tournament selection(4)	Two cuts recombination (0,6)	Binary mutation (0,03)		1147,6	3,826225294
19	200	50	Tournament selection(2)	Two cuts recombination (0,7)	Binary mutation (0,01)		1147,46	3,816857346
20	200	50	Tournament selection(4)	One cut recombination (0,7)	Binary mutation (0,025)		1147,46	3,816857346

3.2 – Population size

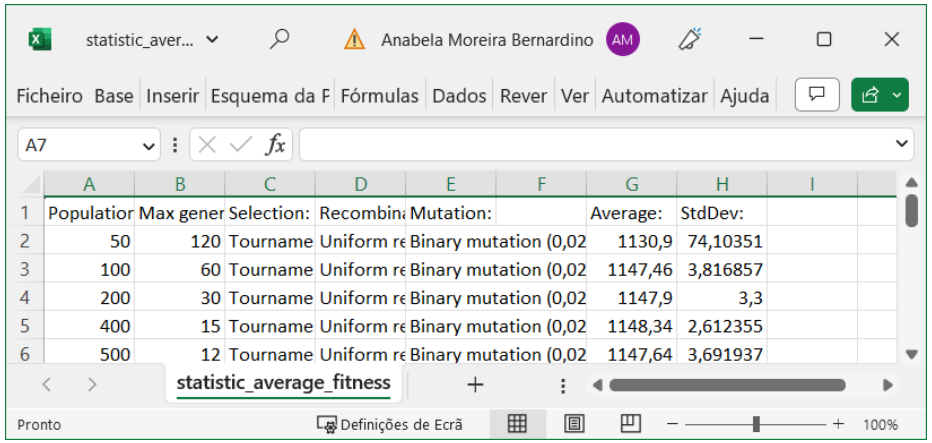
- The number of generations is related to the size of the population and the computational time available for running the algorithm
- The user must choose between using a small population with many generations or a larger population with fewer generations
- Since in the general tests the best result was obtained with a population of 200 individuals and 50 generations, we will test various population sizes, to confirm whether this is indeed the best value:
  - Pop 50 → 120 gerações
  - Pop 100 → 60 gerações
  - Pop 200 → 30 gerações
  - Pop 400 → 15 gerações
  - Pop 500 → 12 gerações
- The config file shown on the right makes combinations of all population types, with all maximum numbers of generations. In the remaining parameters, the best combination obtained in the general tests of Dataset1 is placed. The graph to be created should contain only 5 bars, and the remaining data must be deleted

```
Runs: 50

Population_size: 50, 100, 200, 400, 500

Max_generations: 12, 15, 30, 60, 120

#--------------------------------

Selection: tournament

Tournament_size: 4

#--------------------------------

Recombination: uniform

Recombination_probability: 0.8

#--------------------------------

Mutation: binary

Mutation_probability: 0.025

#--------------------------------

Probability_of_1s: 0.05

Fitness_type: 0

#--------------------------------

Problem_file: ./data_set_1.txt

#--------------------------------

Statistic: BestIndividual
Statistic: BestAverage
```

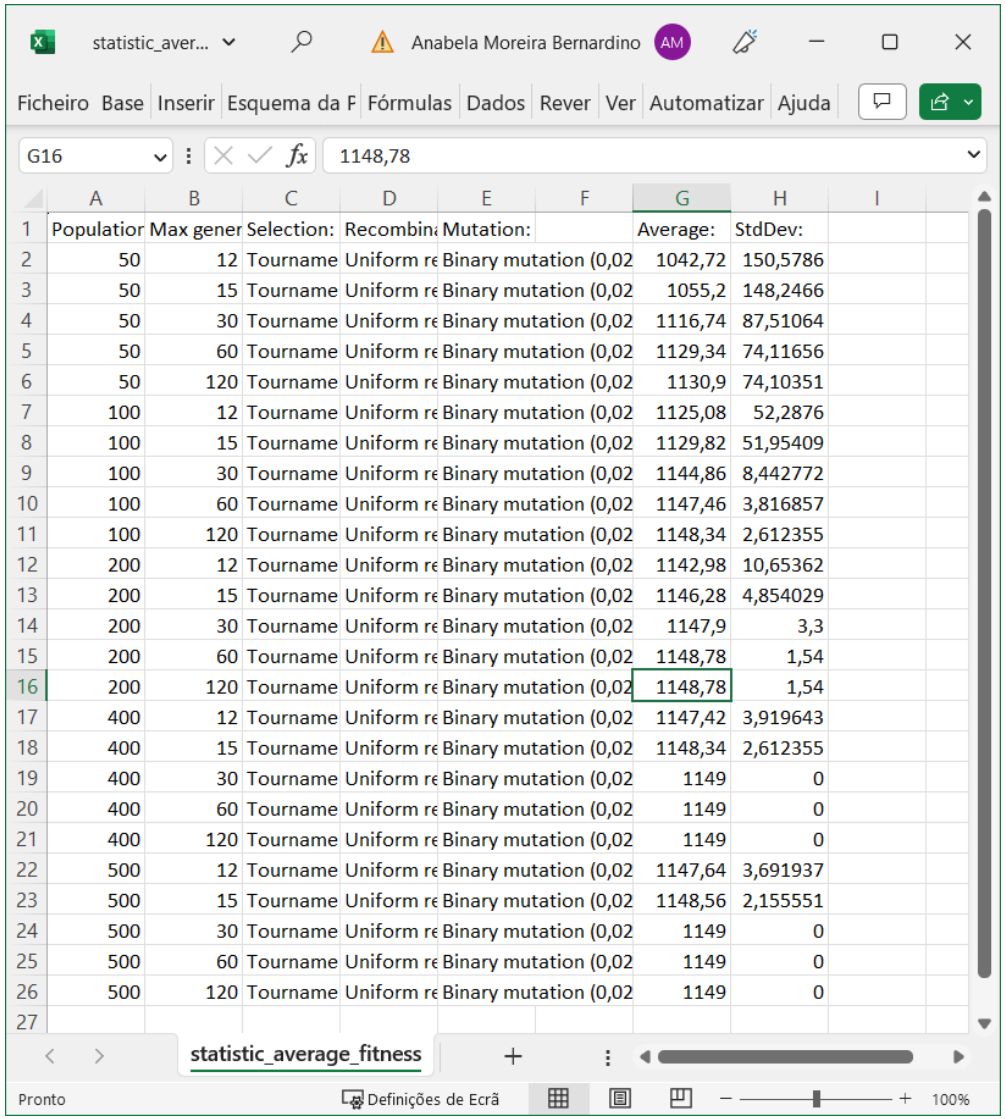
3.2 – Population size (cont.)

- After completing the tests, you should open the *statistic_average_fitness* file

- Delete extra lines:



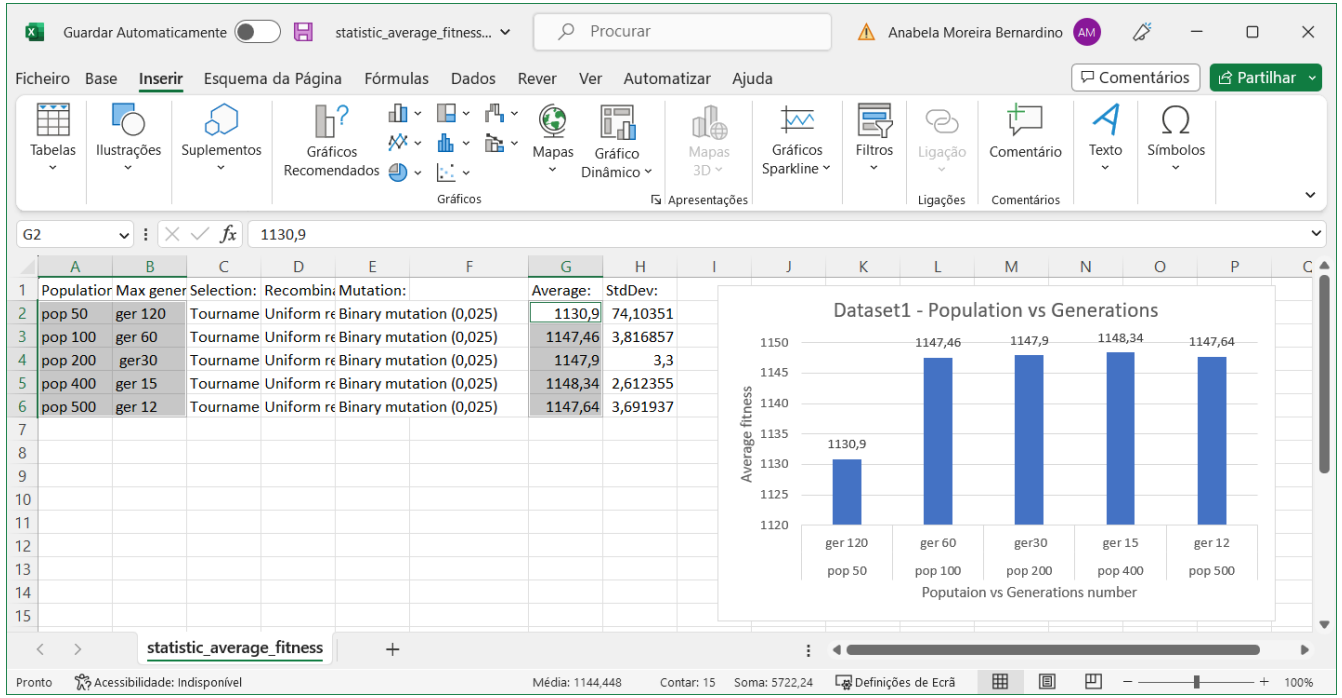
	A	B	C	D	E	F	G	H	I
1	Populatior	Max gener	Selection:	Recombina	Mutation:		Average:	StdDev:	
2	50	120	Tourname	Uniform re	Binary mutation (0,02		1130,9	74,10351	
3	100	60	Tourname	Uniform re	Binary mutation (0,02		1147,46	3,816857	
4	200	30	Tourname	Uniform re	Binary mutation (0,02		1147,9	3,3	
5	400	15	Tourname	Uniform re	Binary mutation (0,02		1148,34	2,612355	
6	500	12	Tourname	Uniform re	Binary mutation (0,02		1147,64	3,691937	



	A	B	C	D	E	F	G	H	I
1	Populatior	Max gener	Selection:	Recombina	Mutation:		Average:	StdDev:	
2	50	12	Tourname	Uniform re	Binary mutation (0,02		1042,72	150,5786	
3	50	15	Tourname	Uniform re	Binary mutation (0,02		1055,2	148,2466	
4	50	30	Tourname	Uniform re	Binary mutation (0,02		1116,74	87,51064	
5	50	60	Tourname	Uniform re	Binary mutation (0,02		1129,34	74,11656	
6	50	120	Tourname	Uniform re	Binary mutation (0,02		1130,9	74,10351	
7	100	12	Tourname	Uniform re	Binary mutation (0,02		1125,08	52,2876	
8	100	15	Tourname	Uniform re	Binary mutation (0,02		1129,82	51,95409	
9	100	30	Tourname	Uniform re	Binary mutation (0,02		1144,86	8,442772	
10	100	60	Tourname	Uniform re	Binary mutation (0,02		1147,46	3,816857	
11	100	120	Tourname	Uniform re	Binary mutation (0,02		1148,34	2,612355	
12	200	12	Tourname	Uniform re	Binary mutation (0,02		1142,98	10,65362	
13	200	15	Tourname	Uniform re	Binary mutation (0,02		1146,28	4,854029	
14	200	30	Tourname	Uniform re	Binary mutation (0,02		1147,9	3,3	
15	200	60	Tourname	Uniform re	Binary mutation (0,02		1148,78	1,54	
16	200	120	Tourname	Uniform re	Binary mutation (0,02		1148,78	1,54	
17	400	12	Tourname	Uniform re	Binary mutation (0,02		1147,42	3,919643	
18	400	15	Tourname	Uniform re	Binary mutation (0,02		1148,34	2,612355	
19	400	30	Tourname	Uniform re	Binary mutation (0,02		1149	0	
20	400	60	Tourname	Uniform re	Binary mutation (0,02		1149	0	
21	400	120	Tourname	Uniform re	Binary mutation (0,02		1149	0	
22	500	12	Tourname	Uniform re	Binary mutation (0,02		1147,64	3,691937	
23	500	15	Tourname	Uniform re	Binary mutation (0,02		1148,56	2,155551	
24	500	30	Tourname	Uniform re	Binary mutation (0,02		1149	0	
25	500	60	Tourname	Uniform re	Binary mutation (0,02		1149	0	
26	500	120	Tourname	Uniform re	Binary mutation (0,02		1149	0	
27									

3.2 – Population size (cont.)

- Creating the chart automatically:
  - Convert the generation and population columns to text (in the example below, the text *pop* was added to the population column and *ger* to the generation column)
  - Convert the average fitness column to a number (if the numbers do not appear aligned to the right, it is necessary to change the formatting of the numbers in the Excel options, or simply select the column data and replace '.' with ',')
  - Select the population size, number of generations and average fitness data columns and create a column chart
  - Add titles on the vertical and horizontal axes and add a title to the chart
  - Add data labels to chart columns



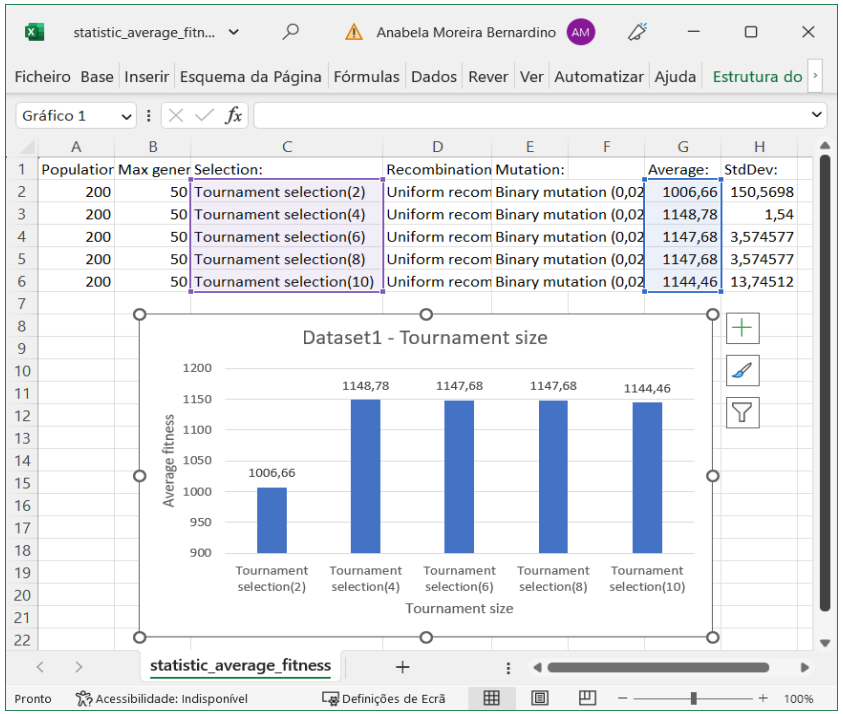
Populatior	Max gener	Selection:	Recombin	Mutation:	Average:	StdDev:
pop 50	ger 120	Tourname	Uniform re	Binary mutation (0,025)	1130,9	74,10351
pop 100	ger 60	Tourname	Uniform re	Binary mutation (0,025)	1147,46	3,816857
pop 200	ger30	Tourname	Uniform re	Binary mutation (0,025)	1147,9	3,3
pop 400	ger 15	Tourname	Uniform re	Binary mutation (0,025)	1148,34	2,612355
pop 500	ger 12	Tourname	Uniform re	Binary mutation (0,025)	1147,64	3,691937

As can be seen from the chart, the populations that show the best results are the 200 and 400

3.3 – Tournament size

- The config file shown on the right contains the tournament size variation
- In the remaining parameters, the best combination obtained in the general tests is placed
- After carrying out the tests, open the *statistic_average_fitness* file and create a chart identical to the one shown below (check before creating the chart that the data in the Average column is aligned to the right)

As can be seen in the graph, the best result was obtained for tournament 4

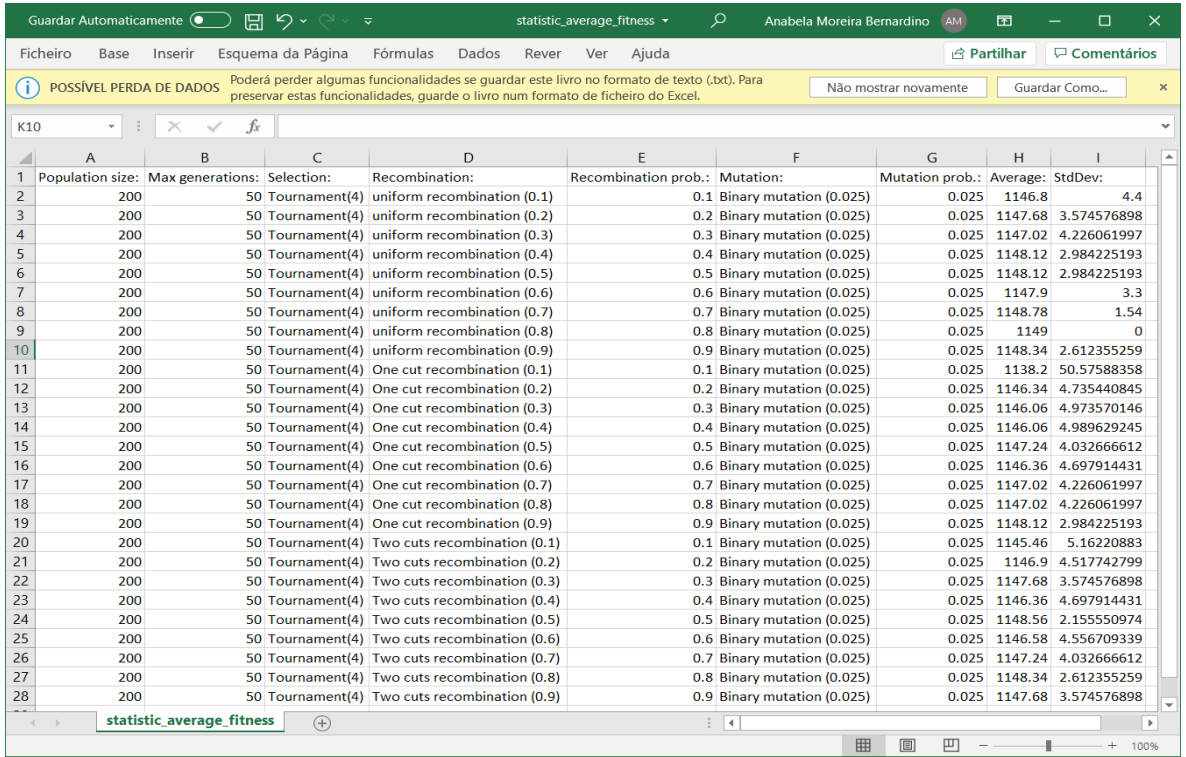


	A	B	C	D	E	F	G	H
1	Populatior	Max gener	Selection:	Recombination	Mutation:		Average:	StdDev:
2	200	50	Tournament selection(2)	Uniform recom	Binary mutation (0,02		1006,66	150,5698
3	200	50	Tournament selection(4)	Uniform recom	Binary mutation (0,02		1148,78	1,54
4	200	50	Tournament selection(6)	Uniform recom	Binary mutation (0,02		1147,68	3,574577
5	200	50	Tournament selection(8)	Uniform recom	Binary mutation (0,02		1147,68	3,574577
6	200	50	Tournament selection(10)	Uniform recom	Binary mutation (0,02		1144,46	13,74512

```
Runs: 50

Population_size: 200

Max_generations: 50

#------------------------------

Selection: tournament

Tournament_size: 2, 4, 6, 8, 10

#------------------------------

Recombination: uniform

Recombination_probability: 0.8

#------------------------------

Mutation: binary

Mutation_probability: 0.025

#------------------------------

Probability_of_1s: 0.05

Fitness_type: 0

#------------------------------

Problem_file: ./data_set_1.txt

#------------------------------

Statistic: BestIndividual
Statistic: BestAverage
```

3.4 – Recombination

- The config file shown on the right contains the variation of the various recombination methods with the various recombination probabilities
- In the remaining parameters, the best combination obtained in the general tests is placed
- After carrying out the tests, the *statistic_average_fitness* file should be opened



	A	B	C	D	E	F	G	H	I
1	Population size:	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	200	50	Tournament(4)	uniform recombination (0.1)	0.1	Binary mutation (0.025)	0.025	1146.8	4.4
3	200	50	Tournament(4)	uniform recombination (0.2)	0.2	Binary mutation (0.025)	0.025	1147.68	3.574576898
4	200	50	Tournament(4)	uniform recombination (0.3)	0.3	Binary mutation (0.025)	0.025	1147.02	4.226061997
5	200	50	Tournament(4)	uniform recombination (0.4)	0.4	Binary mutation (0.025)	0.025	1148.12	2.984225193
6	200	50	Tournament(4)	uniform recombination (0.5)	0.5	Binary mutation (0.025)	0.025	1148.12	2.984225193
7	200	50	Tournament(4)	uniform recombination (0.6)	0.6	Binary mutation (0.025)	0.025	1147.9	3.3
8	200	50	Tournament(4)	uniform recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1148.78	1.54
9	200	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
10	200	50	Tournament(4)	uniform recombination (0.9)	0.9	Binary mutation (0.025)	0.025	1148.34	2.612355259
11	200	50	Tournament(4)	One cut recombination (0.1)	0.1	Binary mutation (0.025)	0.025	1138.2	50.57588358
12	200	50	Tournament(4)	One cut recombination (0.2)	0.2	Binary mutation (0.025)	0.025	1146.34	4.735440845
13	200	50	Tournament(4)	One cut recombination (0.3)	0.3	Binary mutation (0.025)	0.025	1146.06	4.973570146
14	200	50	Tournament(4)	One cut recombination (0.4)	0.4	Binary mutation (0.025)	0.025	1146.06	4.989629245
15	200	50	Tournament(4)	One cut recombination (0.5)	0.5	Binary mutation (0.025)	0.025	1147.24	4.032666612
16	200	50	Tournament(4)	One cut recombination (0.6)	0.6	Binary mutation (0.025)	0.025	1146.36	4.697914431
17	200	50	Tournament(4)	One cut recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.02	4.226061997
18	200	50	Tournament(4)	One cut recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1147.02	4.226061997
19	200	50	Tournament(4)	One cut recombination (0.9)	0.9	Binary mutation (0.025)	0.025	1148.12	2.984225193
20	200	50	Tournament(4)	Two cuts recombination (0.1)	0.1	Binary mutation (0.025)	0.025	1145.46	5.16220883
21	200	50	Tournament(4)	Two cuts recombination (0.2)	0.2	Binary mutation (0.025)	0.025	1146.9	4.517742799
22	200	50	Tournament(4)	Two cuts recombination (0.3)	0.3	Binary mutation (0.025)	0.025	1147.68	3.574576898
23	200	50	Tournament(4)	Two cuts recombination (0.4)	0.4	Binary mutation (0.025)	0.025	1146.36	4.697914431
24	200	50	Tournament(4)	Two cuts recombination (0.5)	0.5	Binary mutation (0.025)	0.025	1148.56	2.155550974
25	200	50	Tournament(4)	Two cuts recombination (0.6)	0.6	Binary mutation (0.025)	0.025	1146.58	4.556709339
26	200	50	Tournament(4)	Two cuts recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.24	4.032666612
27	200	50	Tournament(4)	Two cuts recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.34	2.612355259
28	200	50	Tournament(4)	Two cuts recombination (0.9)	0.9	Binary mutation (0.025)	0.025	1147.68	3.574576898

Runs: 50

Population_size: 200

Max_generations: 50

#--------------------------------

Selection: tournament

Tournament_size: 4

#--------------------------------

Recombination: uniform, one_cut, two_cuts

Recombination_probability: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

#--------------------------------

Mutation: binary

Mutation_probability: 0.025

#--------------------------------

Probability_of_1s: 0.05

Fitness_type: 0

#--------------------------------

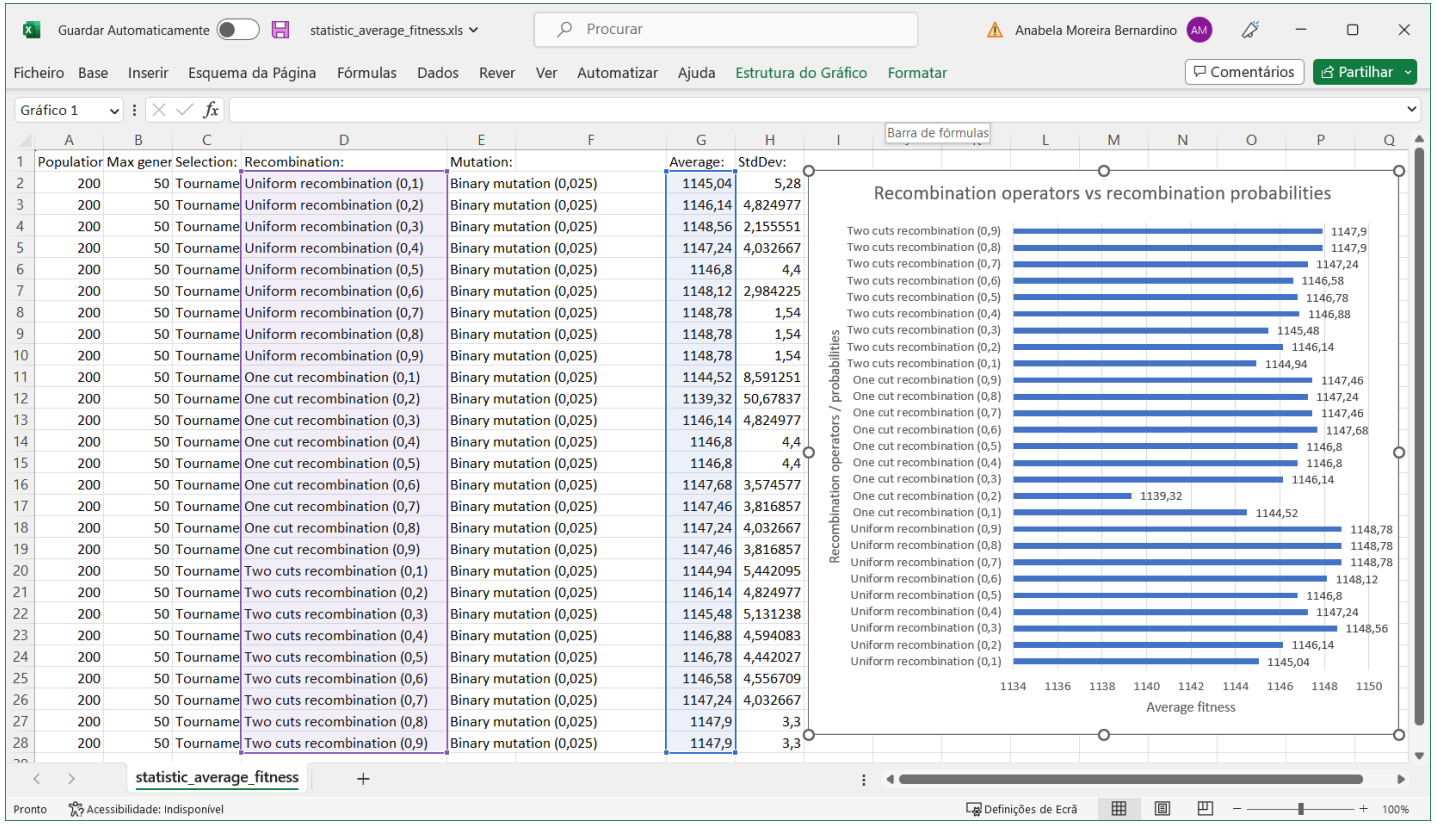
Problem_file: ./data_set_1.txt

#--------------------------------

Statistic: BestIndividual
Statistic: BestAverage

3.4 – Recombination (cont.)

- Check that the data in the Average column is aligned to the right (in the example below, the '.' was replaced by ',' since the data in this column was recognized as a number)
- Select data from Recombination and Average columns and create a bar chart
- Increase the size of the chart vertically so that all probabilities from 0.1 to 0.9 appear

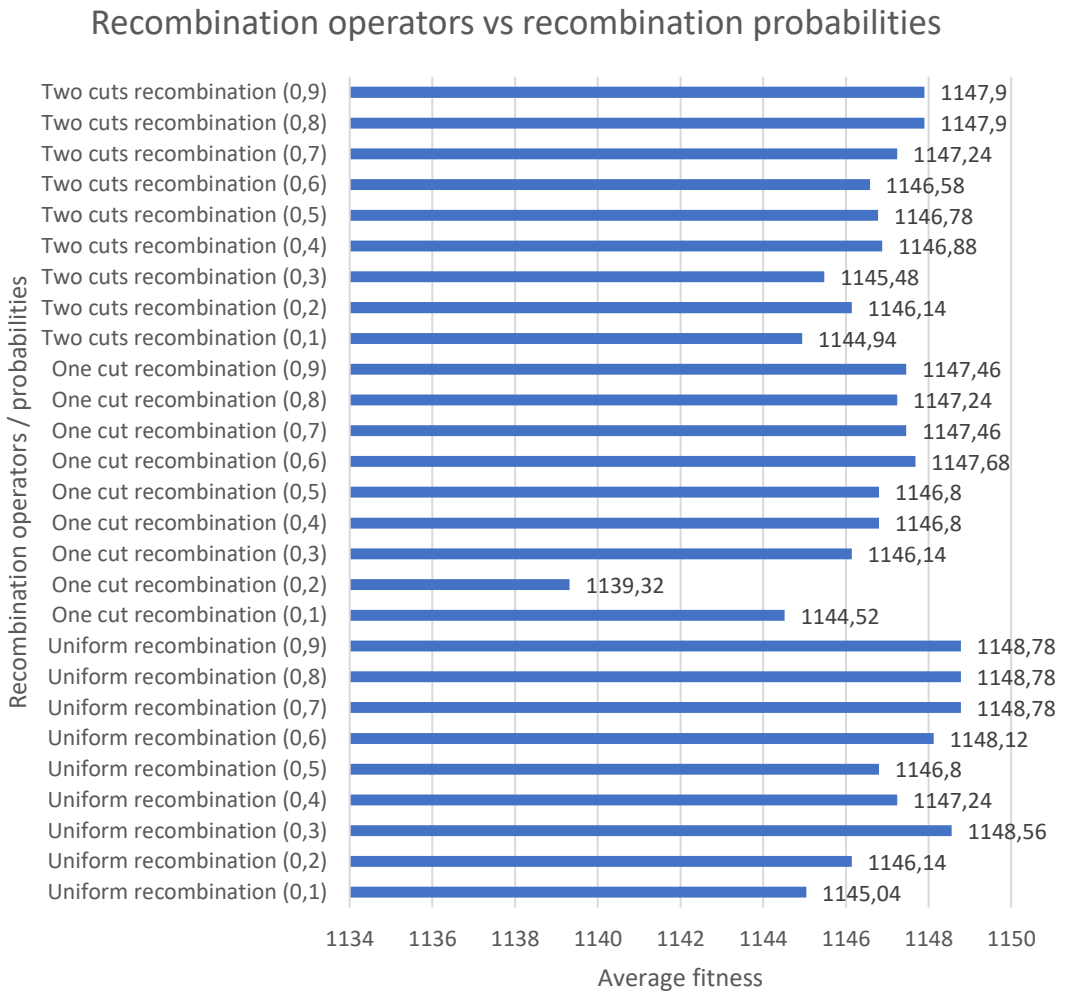


	A	B	C	D	E	F	G	H
1	Populatior	Max gener	Selection:	Recombination:	Mutation:		Average:	StdDev:
2	200	50	Tourname	Uniform recombination (0,1)	Binary mutation (0,025)		1145,04	5,28
3	200	50	Tourname	Uniform recombination (0,2)	Binary mutation (0,025)		1146,14	4,824977
4	200	50	Tourname	Uniform recombination (0,3)	Binary mutation (0,025)		1148,56	2,155551
5	200	50	Tourname	Uniform recombination (0,4)	Binary mutation (0,025)		1147,24	4,032667
6	200	50	Tourname	Uniform recombination (0,5)	Binary mutation (0,025)		1146,8	4,4
7	200	50	Tourname	Uniform recombination (0,6)	Binary mutation (0,025)		1148,12	2,984225
8	200	50	Tourname	Uniform recombination (0,7)	Binary mutation (0,025)		1148,78	1,54
9	200	50	Tourname	Uniform recombination (0,8)	Binary mutation (0,025)		1148,78	1,54
10	200	50	Tourname	Uniform recombination (0,9)	Binary mutation (0,025)		1148,78	1,54
11	200	50	Tourname	One cut recombination (0,1)	Binary mutation (0,025)		1144,52	8,591251
12	200	50	Tourname	One cut recombination (0,2)	Binary mutation (0,025)		1139,32	50,67837
13	200	50	Tourname	One cut recombination (0,3)	Binary mutation (0,025)		1146,14	4,824977
14	200	50	Tourname	One cut recombination (0,4)	Binary mutation (0,025)		1146,8	4,4
15	200	50	Tourname	One cut recombination (0,5)	Binary mutation (0,025)		1146,8	4,4
16	200	50	Tourname	One cut recombination (0,6)	Binary mutation (0,025)		1147,68	3,574577
17	200	50	Tourname	One cut recombination (0,7)	Binary mutation (0,025)		1147,46	3,816857
18	200	50	Tourname	One cut recombination (0,8)	Binary mutation (0,025)		1147,24	4,032667
19	200	50	Tourname	One cut recombination (0,9)	Binary mutation (0,025)		1147,46	3,816857
20	200	50	Tourname	Two cuts recombination (0,1)	Binary mutation (0,025)		1144,94	5,442095
21	200	50	Tourname	Two cuts recombination (0,2)	Binary mutation (0,025)		1146,14	4,824977
22	200	50	Tourname	Two cuts recombination (0,3)	Binary mutation (0,025)		1145,48	5,131238
23	200	50	Tourname	Two cuts recombination (0,4)	Binary mutation (0,025)		1146,88	4,594083
24	200	50	Tourname	Two cuts recombination (0,5)	Binary mutation (0,025)		1146,78	4,442027
25	200	50	Tourname	Two cuts recombination (0,6)	Binary mutation (0,025)		1146,58	4,556709
26	200	50	Tourname	Two cuts recombination (0,7)	Binary mutation (0,025)		1147,24	4,032667
27	200	50	Tourname	Two cuts recombination (0,8)	Binary mutation (0,025)		1147,9	3,3
28	200	50	Tourname	Two cuts recombination (0,9)	Binary mutation (0,025)		1147,9	3,3

3.4 – Recombination (cont.)

- After creating the chart, add titles on the vertical and horizontal axes and add a title to the chart
- Add data labels to chart columns

Observing the chart, we can conclude:
- The best result is obtained for uniform recombination, probabilities 0.7, 0.8 and 0.9
- The best probabilities for two-cut recombination are 0.8 and 0.9 and for one-cut recombination is 0.6



3.5 – Mutation

- In the Knapsack project there was only one mutation implemented, so it was only necessary to vary the various mutation probabilities

- In the project to be developed at the AI course, a config file identical to the one created for recombination should be created and a chart showing the influence of the various probabilities for each mutation method should be created

Performing experiments for the course project

Considerations for automated experiments:

- Use between 30 to 50 runs for each combination of parameters
- Save the config files created for each experiment and the files resulting from the experiments (which must be delivered together with the project code)
- After carrying out all the experiments, delete all .exe files from the test folders

The project report should include:

- Information on the best combination of parameters for each dataset
- The best result obtained (best fitness) and the fitness average (of the *x* runs performed)
- A chart with the results obtained for each combination of parameters for each dataset (tournament, population vs generations, recombinations, mutations)
- All results presented in tables and charts must be properly described in the text of the report

Have a nice time performing experiments ☺