EINDHOVEN UNIVERSITY OF TECHNOLOGY

Assignment 1
Evolutionary Process Discovery

1BM120 group 6

May 14, 2024

Name	ID
Alicja Obszyńska	1505084
Arthur Hafkenscheid	1452401
Carleijn den Brok	2036274
Hidde Huitema	1373005
Georgios Parlitsis	2026759

Contents

1 Question 1

Figure 1.1 shows the best fitness for each generation of the GA for a population size of 50 and 100 generations. This GA was made with the configuration shown in Table 1.1. Figure 1.1 shows that the GA quickly finds a (local) optimal solution to the problem.

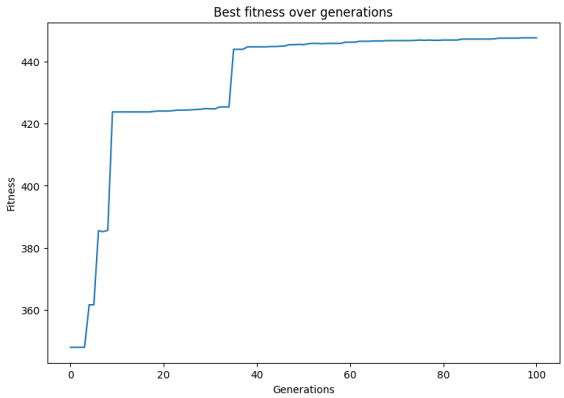

Figure 1.1: Best fitness for each generation

config	crossover	p_{cross}	mutation	p_{mut}	selection	Best fitness	time [s]
1	cxTwoPoint	0.20	mutFlipBit	0.20	selTournament	447.7	8.7

Table 1.1: Configurations for question 1

2 Question 2

For a population size of 50 and 100 generations, Figure 2.1 is obtained. This image shows the average best fitness of each configuration at each generation. This plot is obtained by running each configuration 30 times and averaging the best fitness for each generation over these 30 runs. Table 2.1 shows the different configurations that were tried for the GA.

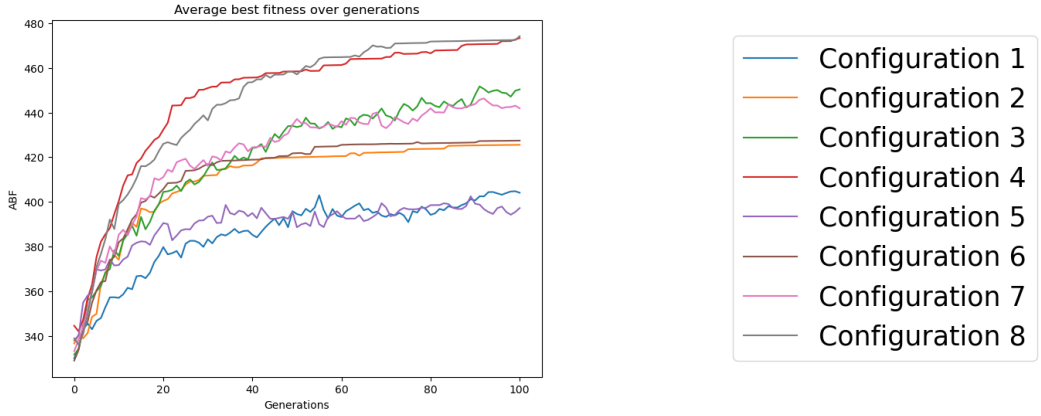

Figure 2.1: Average best fitness for each generation for each configuration.

Next, Figure 2.2 is the corresponding boxplot for the best individuals for each configuration.

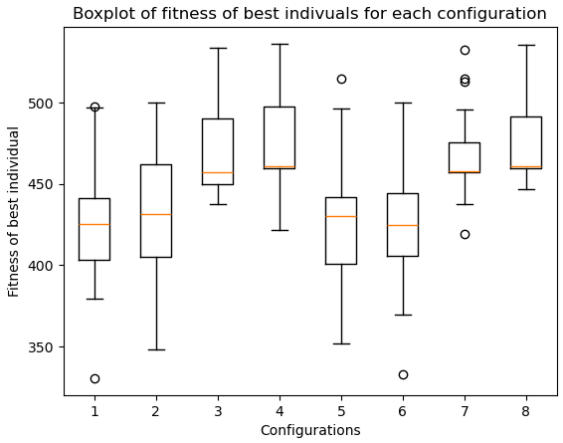

Figure 2.2: Boxplot with the best individuals for each configuration.

The different configurations can be seen in Table 2.1. This table also shows the Average Best Fitness (ABF) value found by each configuration as well as the time it takes to run 30 iterations of each of the configurations.

config	crossover	p_{cross}	mutation	p_{mut}	selection	ABF	time [s]
1	cxTwoPoint	0.20	mutFlipBit	0.20	selRoulette	404.76	45
2	cxTwoPoint	0.20	mutFlipBit	0.20	selTournament	425.55	48
3	cxTwoPoint	0.20	mutShuffleIndexes	0.20	selRoulette	451.71	47
4	cxTwoPoint	0.20	mutShuffleIndexes	0.20	selTournament	473.36	51
5	cxOrdered	0.20	mutFlipBit	0.20	selRoulette	402.49	48
6	cxOrdered	0.20	mutFlipBit	0.20	selTournament	427.43	52
7	cxOrdered	0.20	mutShuffleIndexes	0.20	selRoulette	446.32	48
8	cxOrdered	0.20	mutShuffleIndexes	0.20	selTournament	474.21	53

Table 2.1: Configurations for Question 2.

Therefore, the settings that will be used in the subsequent exercises are cxOrdered and mutShuffleIndexes with selTournament, since this configuration managed to find the individual with the best overall fitness. This configuration also takes the longest to run, but the time difference between different configurations is relatively small so this slightly longer running time creates no objection against using this configuration.

3 Question 3

For a population size of 50 and 100 generations, Figure 3.1 is obtained. This image shows the average best fitness of each configuration. The sixteen different configurations are tabulated in Table 3.1 alongside the resulting ABF and running time.

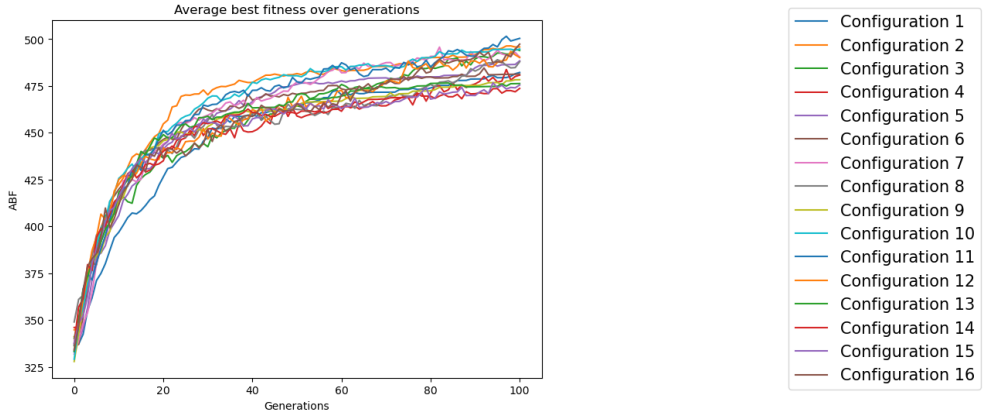

Figure 3.1: Average best fitness for each generation for each configuration.

Next, Figure 3.2 is the corresponding boxplot for the best individuals for each configuration.

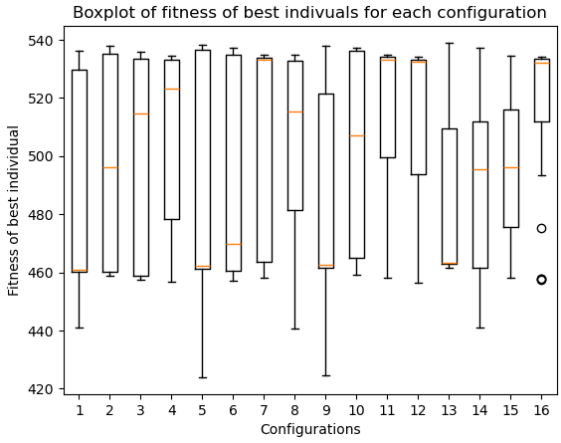

Figure 3.2: Boxplot with the best individuals for each configuration.

ABF		Crossover probabilities			
		0.2	0.4	0.6	0.8
mutation probabilities	0.2	481.03	485.09	484.02	490.49
	0.4	491.36	491.22	486.09	491.63
	0.6	491.84	476.77	485.21	492.36
	0.8	483.41	475.75	498.05	497.17

Table 3.1: Average best fitness at final iteration for each configuration.

Therefore, the best settings for the GA for this problem are cxOrdered with a probability of 0.6, and mutShuffleIndexes with a probability of 0.8.

4 Question 4

A decorator is a wrapper that is called instead of a function and it can be applied after mutation or crossover. In our case it ensures that we have at most two nonzero entries in each column. It checks which ones are nonzero and randomly chooses the indexes of the entries to be kept in each column while the remaining entries are set to zero.

The decorator can be seen in the file with the code.

5 Question 5

The best Petri net determined in exercise 3 was the one with configuration 12 - with crossover probability of 0.6 and mutation probability of 0.8, which can be found in Table 3.1.

The original network and the one after using the decorator can be seen in figures 5.1 and 5.2. It is visible that the nodes in the network before the decorator (Figure 5.1) are all linked, which means that it is overfitted and does not tell us anything about specific underlying connections. The other network, on the other hand, has much fewer connections which suggests that it is underfitted. It would mean that not all the actual connections can be seen there and that is also not desirable.

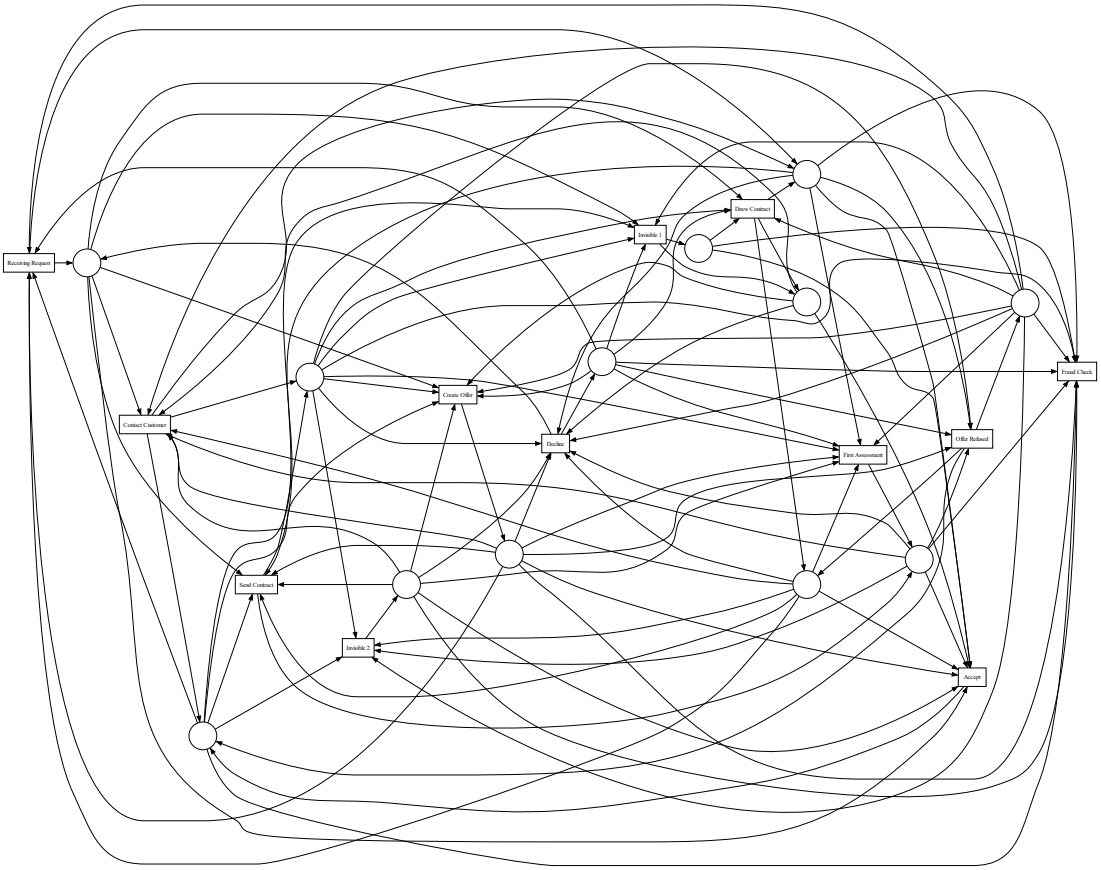

Figure 5.1: Unconstrained Petri net.

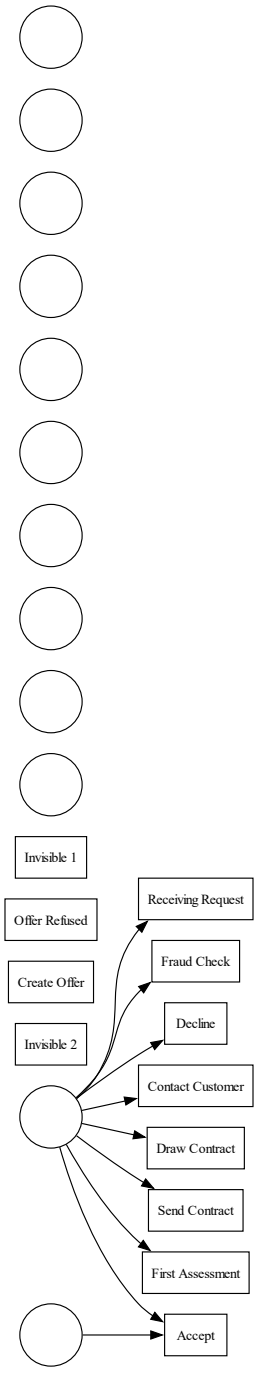

Figure 5.2: Sparse Petri net.