Name: Serban Bogdan Alexandru

ID: 1856482

Lab 3

- **Exercise 4**

Algorithm 1 Genetic algorithm

Require: Population size: $\lambda \in \mathbb{N}$
Require: Mutation rate: $\chi \in \mathbb{R}$
Require: Crossover ratio: $\alpha \in (0,1)$
Require: Tournament size: k $\in \mathbb{N}$
Require: Maximum generations: maxT $\in \mathbb{N}$
Require: Number of weeks: weeks $\in \mathbb{N}$
Require: Strategy size: h $\in \mathbb{N}$

for $i = 1$ to λ **do**
  for $j = 1$ to h **do**
    $p[j] \leftarrow Unif(0,1)$
    $A[j] \leftarrow$ Random probability distribution of size **h**
    $B[j] \leftarrow$ Random probability distribution of size **h**
  $P_0[i] \leftarrow (p, a, b)$
$fitness_0 \leftarrow$ individual payoff after simulation of bar attendance
for $t = 1$ to $maxT$ **do**
  for $i = 1$ to λ **do**
    $x \leftarrow tournament\_selection(P_t)$
    $y \leftarrow tournament\_selection(P_t)$
    $offsprings_t(i) \leftarrow mutation(intermediate\_crossover(x, y))$
  $offspring\_fitness_t \leftarrow$ individual payoff after simulation of bar attendance
  $P_{t+1} \leftarrow$ best λ individuals from P_t and $offsprings_t$
  $fitness_t \leftarrow$ individual payoff after simulation of bar attendance
return P_t

Algorithm 3 Crossover

Require: Strategy size: $h \in \mathbb{N}$
Require: Crossover ratio: $\alpha \in (0,1)$
Require: First Parent: $parent_1 = [p_1, A_1, B_1]$ where $p_1 \in (0,1)^h; A_1, B_1 \in (0,1)^h \times (0,1)^h$
Require: First Parent: $parent_2 = [p_2, A_2, B_2]$ where $p_2 \in (0,1)^h; A_2, B_2 \in (0,1)^h \times (0,1)^h$

$p_{offspring}[i] \leftarrow \alpha * p_1[i] + (1-\alpha) * p_2[i], i = \overline{1 \ldots h}$
$A_{offspring}[i][j] \leftarrow \alpha * A_1[i][j] + (1-\alpha) * A_2[i][j], i, j = \overline{1 \ldots h}$
$B_{offspring}[i][j] \leftarrow \alpha * B_1[i][j] + (1-\alpha) * B_2[i][j], i, j = \overline{1 \ldots h}$
return $[p_{offspring}, A_{offspring}, B_{offspring}]$

Algorithm 2 Mutation operator

Require: Mutation rate: $\chi \in \mathbb{R}$
Require: Strategy size: $h \in \mathbb{N}$
Require: Individual to mutate: strategy $= [p, A, B]$ where $p \in (0,1)^h$,$A, B \in (0,1)^h \times (0,1)^h$

for $i = 1$ to h **do**
    prob $\sim Unif(0,1)$
    if $prob \leq \chi$ **then**
        p(i) $\leftarrow p(i) + Gauss(0, 0.15)$
for $i = 1$ to h **do**
    for $j = 1$ to h **do**
        prob $\sim Unif(0,1)$
        if $prob \leq \chi$ **then**
            $A[i][j] \leftarrow A[i][j] + Gauss(0, 0.15)$
        prob $\sim Unif(0,1)$
        if $prob \leq \chi$ **then**
            $B[i][j] \leftarrow A[i][j] + Gauss(0, 0.15)$
return strategy

Algorithm 4 Tournament selection

Require: Population size: $\lambda \in \mathbb{N}$
Require: Tournament size: k $\in \mathbb{N}$
Require: Populaton: $P = (y^{(1)}, y^{(2)}, ..., y^{(\lambda)})$ *with* $y^{(i)} = [p_i, A_i, B_i]$ where $p_i \in (0,1)^h; A_i, B_i \in (0,1)^h \times (0,1)^h$
Require: Population fitnesses: $fit_P \in \mathbb{N}^\lambda$

1. Sample uniformly at random (with replacement) a set S of k elements from $\{1, ..., \lambda\}$
2. j $\sim$ Unif($argmax_{i \in S}\ fif_P$)
3. return $y^{(j)}$

Name: Serban Bogdan Alexandru

ID: 1856482

- **Exercise 5**

The purpose of this laboratory was to design and study an evolutionary algorithm that uses coevolution in order to find feasible solutions to the **El Farol Bar** problem, namely to determine how a fixed amount of individuals can efficiently share a limited resource in order to get the maximum payoff. For the setup of the experiments, individuals were fitted with strategies consisting in 15 states, the simulation of each strategy was done across a time period of 15 weeks, and each time, the algorithm was run for 100 generations (due to limited computational resources).

- **Experiment 1: Average weekly attendance vs selection method**

The purpose of this experiment was to determine which selection method is more appropriate for finding solutions to the bar problem. Therefore, the average weekly attendance of the 100th generation, was plotted for each selection method that was studied. The following parameter setting was used:

  - **Population size: n** = 200;
  - **Mutation rate: χ** = 1.5;
  - **Crossover ratio: r** = 0.5.

The selection methods that have been studied during this experiment are **Fitness Proportionate Selection (FPS)** and **Tournament Selection (TS).** A tournament size of 3 was used to conduct this experiment. The genetic algorithm was run 100 times for each selection method and the average weekly attendance was recorded. The results are plotted below:

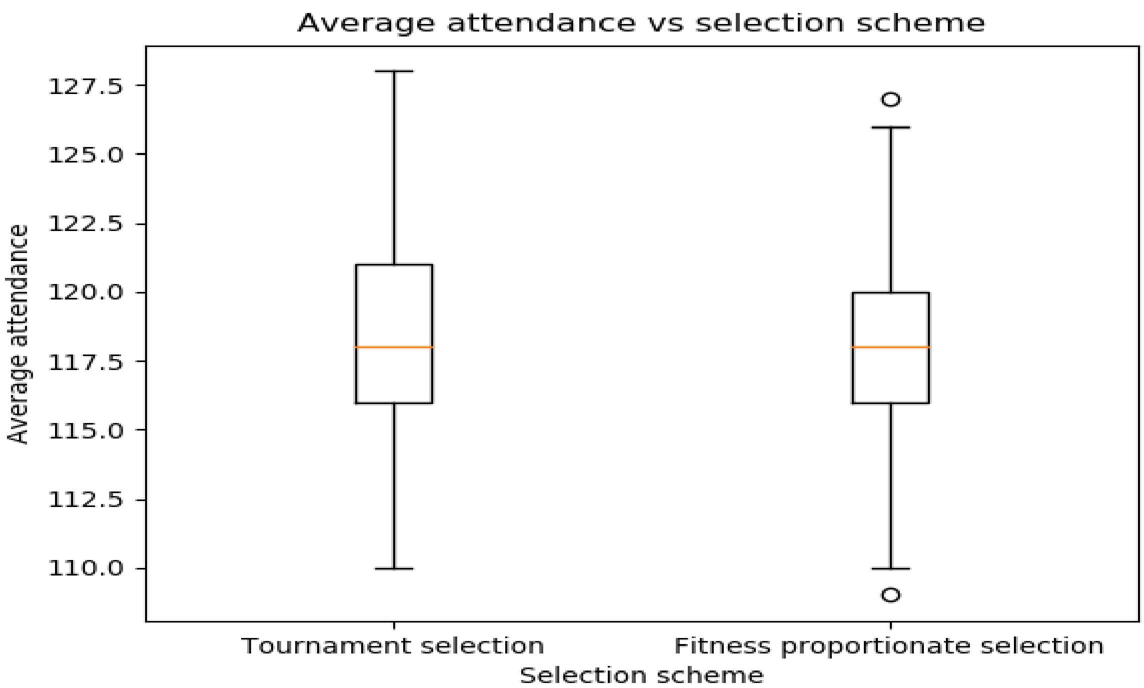

Figure 1 Average attendance vs selection scheme

As it can be seen in **Figure 1,** both selection schemes gave almost similar results. The amplitude of the results for **FPS** tends to be smaller than the one of **TS,** namely ranging between an average attendance of 110 and 126 individuals out of 200, compared to a range of the weekly attendance between 110 and 128 individuals in the case of **TS.** Also, there can be seen that in the case of **FPS,** in 75% of the runs, the attendance at the bar was equal or smaller to the maximum amount of 120 (60% of the total population), and in 50% of the runs, the average weekly attendance at the bar is ranging between 116 and 120 individuals, which means that the on average, the resource was used with maximum efficiency. In the case of **TS,** 75% of the runs generated a population fitted with strategies that gave an average weekly attendance of 121 or fewer individuals, and 50% of the runs generated an average weekly attendance ranging between 114 and 121 individuals.

As both selection schemes performed similarly, I have decided to conduct another experiment in which I have tested whether the tournament size affects the performance of the tournament selection scheme.

- **Experiment 2: Average weekly attendance vs tournament size**

As stated above, the purpose of this experiment is to determine if the performance of the tournament selection scheme can be improved by changing the size of the tournament. The parameter setting used for this experiment is shown below:

- **Population size: n =** 200;
- **Mutation rate:** $\boldsymbol{\chi}$ = 1.5;
- **Crossover ratio: r =** 0.5;
- **Selection scheme:** Tournament Selection;
- **Tournament size:** $2 \leq \mathbf{k} \leq 5$.

For this experiment, I have decided to test tournament size values ranging from 2 to 5 individuals. The algorithm was run 100 times for each tournament size and the average weekly attendance for each run was recorded. The results are shown in the boxplots in **Figure 2.**

As it can be seen, changing the size of the tournament for the tournament selection scheme does not significantly affect the result of the algorithm. It can be seen that the values of the average attendance at the bar range in similar intervals, with slightly lower values for a tournament size of 2 individuals, ranging from 112 individuals, and up to 126, with 75% of the runs having an average attendance of less than 120 individuals, therefore less than the 60% threshold of occupancy. For a tournament of size 3, the weekly attendance ranges in a slightly larger interval, of between 111 and 128 individuals. For tournaments of sizes 4 and 5, the results range in the same intervals. The difference is in the position of the median value, meaning that 50% of the runs had an attendance of fewer than 119 individuals in the case of a tournament size of 4 individuals and less than 116 for a tournament size of 5 individuals.

Therefore, I have decided that a tournament selection with a tournament size of 2 individuals will be used in the final version of the algorithm.

Name: Serban Bogdan Alexandru

ID: 1856482

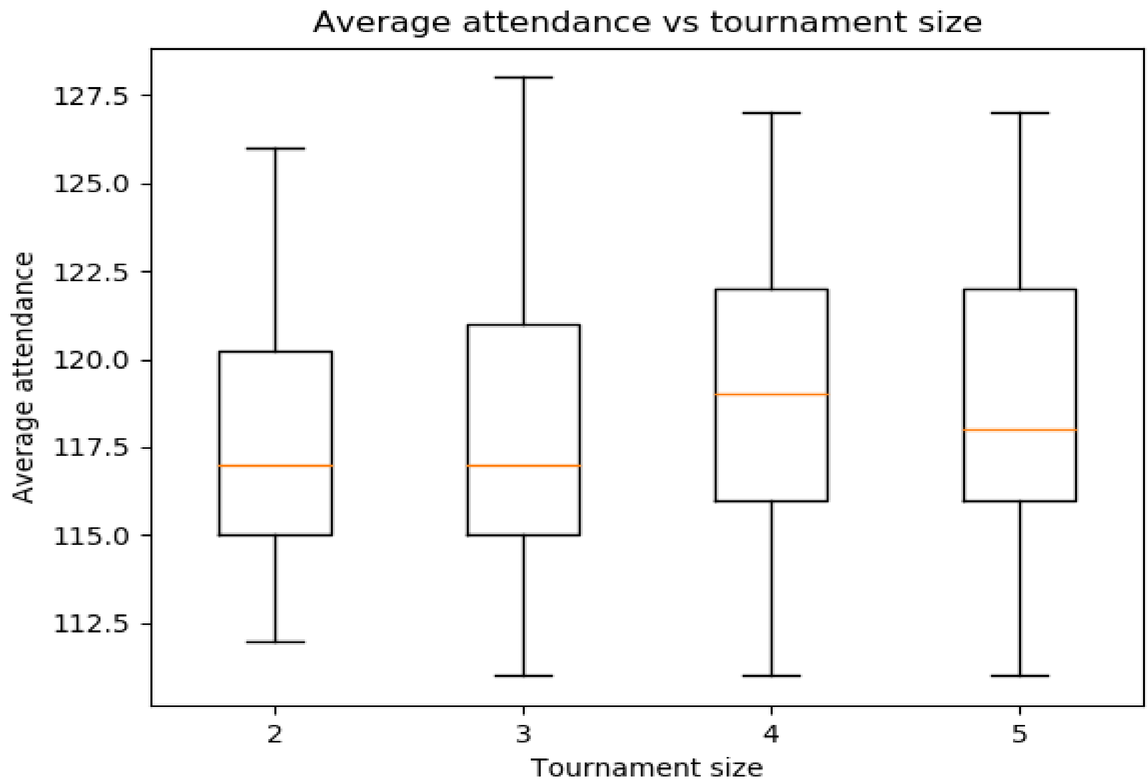

Figure 2 Average attendance vs tournament size

- **Experiment 3: Average attendance vs mutation rate**

In this experiment, I have studied if and how changing values of the mutation rate can determine changes in the average weekly attendance of the individuals to the bar. The following parameter setting was used for this experiment:

- **Population size: n** = 200;
- **Crossover ratio: r** = 0.5;
- **Selection scheme:** Tournament Selection;
- **Tournament size: k** = 3;
- **Mutation rate:** $0.4 \leq \chi \leq 2.8$.

For this experiment, I have decided to use values of the mutation rate ranging from 0.4 up to 2.8, with an incrementing step of 0.4. The genetic algorithm was run 100 times for each of the values of the mutation rate, and the average weekly attendance was recorded. The results are plotted in the boxplots below, in **Figure 3.**

The first aspect that can be noticed while analysing the boxplots is the fact that the runs for all values of the mutation rate except 0.4 returned almost similar results. In the case of a mutation rate of 0.4, the average attendance ranges between 112 and 128 individuals, with 50% of the runs having an attendance of less than 60% of the total population size.

In the case of the other values that have been evaluated, we can observe that for each one of them, 75% of the runs gave solutions which ensured an average attendance of 120 or fewer individuals at the bar. The main difference is in how widely distributed are the attendances resulted from other 25% of the runs. Therefore, we can see that the smaller variation in those runs is obtained by using a mutation rate of 1.2, which ensures that the attendance resulted in that 25 % of the runs range between 120 and 125 individuals, with a small number of extreme values, namely 126, 127 and 128.

Another interesting aspect that can be noticed in the boxplots below is the fact that for a mutation rate of 2.4, the median of the data is set higher than the median of the other runs with different values. That means that in the case of using a mutation rate of 2.4, 50% of the values for the average attendance will be 118 or less, and 25% of the runs will give attendance values between 118 and 120 individuals.

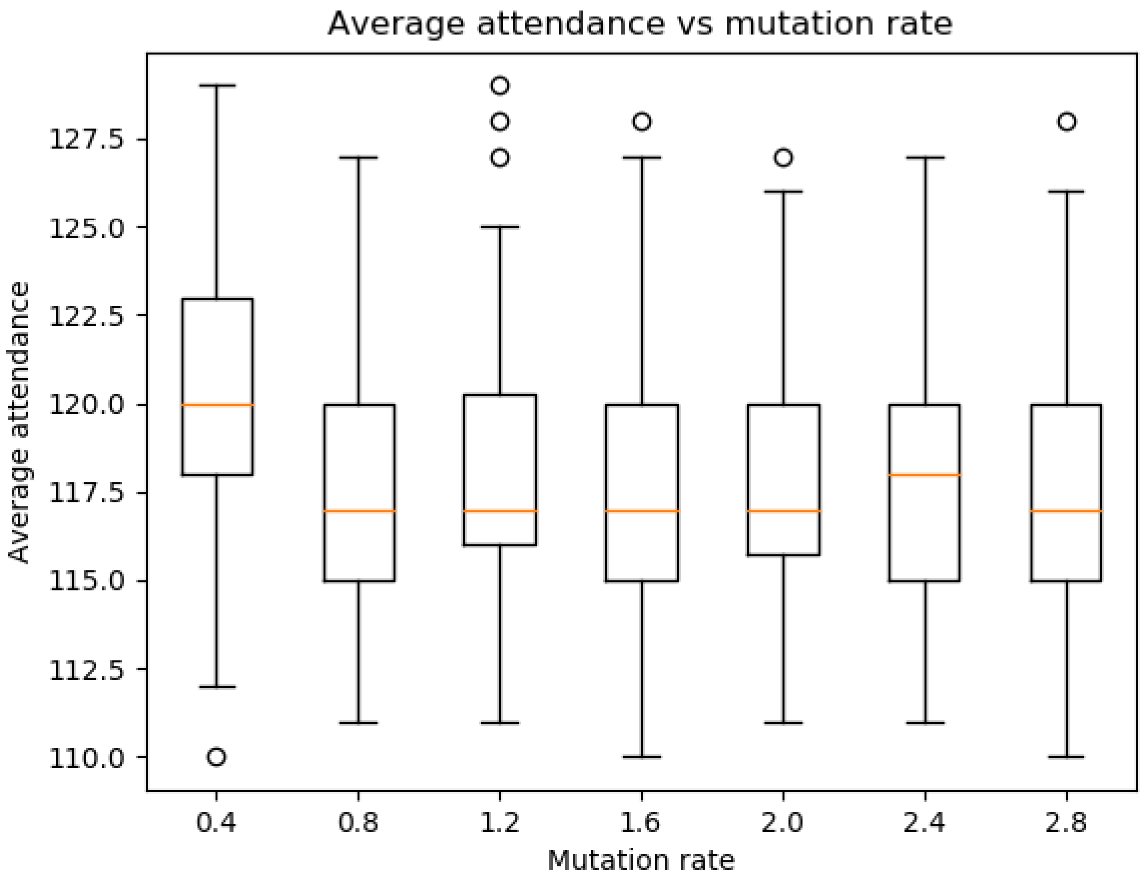

Figure 3 Average attendance vs mutation rate

- **Experiment 4: Average attendance vs crossover ratio**

In this experiment, I have tried to determine a value for the crossover rate that will give the maximum performance of the algorithm. As a reminder, the crossover operator used in building this algorithm is doing a linear combination of individual values of the chromosomes of the parents following an equation which has the following shape:

$$c_i = r * a_i + (1 - r) * b_i,$$

where $\boldsymbol{c_i}$ the chromosome of the offspring is, $\boldsymbol{a_i}$ is the chromosome of the first parent, $\boldsymbol{b_i}$is the chromosome of the second parent and **r** is the crossover ratio. The parameter setting used in this experiment is the following:

- **Population size: n** = 200;

- **Mutation rate:** χ = 1.5;
- **Selection scheme:** Tournament Selection;
- **Tournament size:** **k** = 3;
- **Crossover ratio:** $0.1 \leq \mathbf{r} \leq 0.9$.

For this experiment, I have decided to test crossover ratio values ranging between 0.1 and 0.9, with an increasing step of 0.1. The algorithm was run 100 times for each value and the average bar attendance for each run was recorded. The results are plotted in the boxplots below.

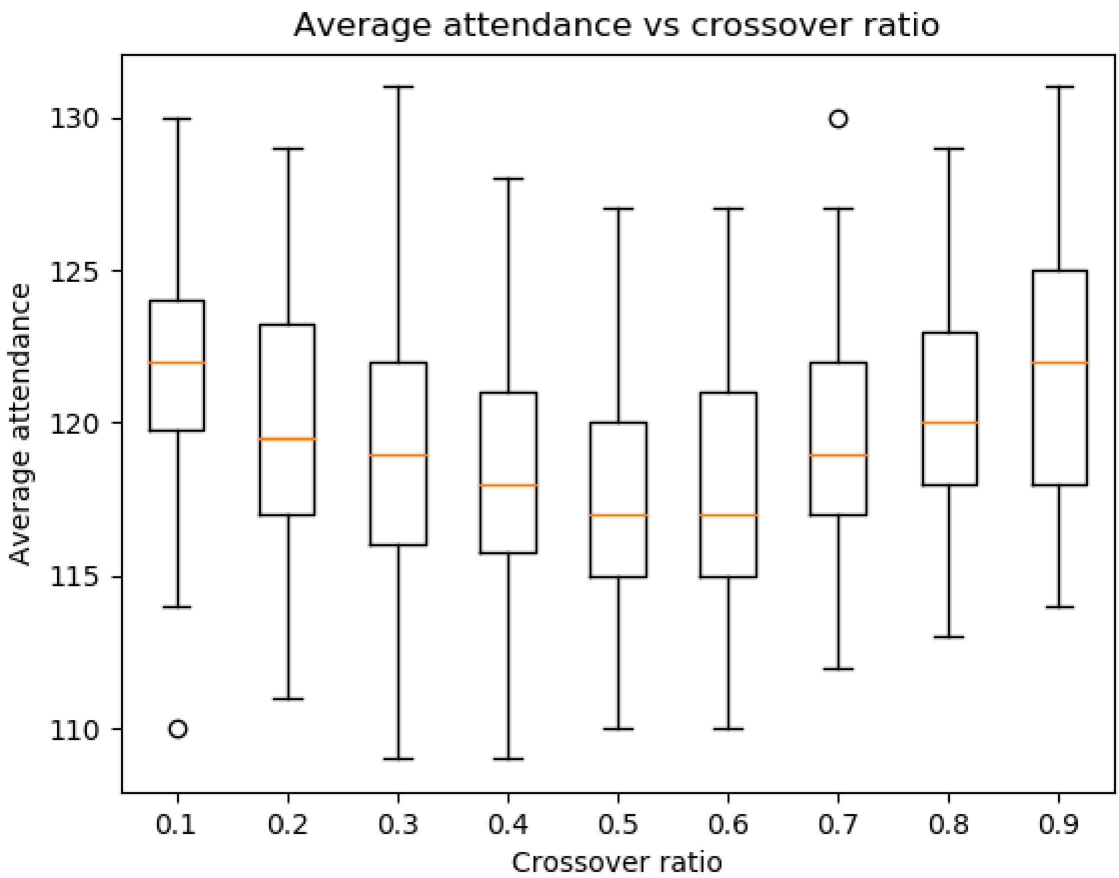

Figure 4 Average attendance vs crossover ratio

Analysing the boxplots in **Figure 4,** the first aspect that can be noticed is the fact that the best performance of the algorithm was recorded for a crossover rate value of 0.5, meaning that when two chromosomes are combined, the resulted offspring will be the average of corresponding genes in the parents. Therefore, it can be seen that for an averaging the chromosomes of the parents, after running the algorithm, in 75% of the runs the average weekly attendance was of 120 individuals (or 60% of the total population) or less.

Looking at other values for the crossover ratio, it can observe that as the values get closer to 0.5, the average attendance outputted by the algorithm converges towards the one outputted by using a value of 0.5.

Given the results of this experiment, I have taken the decision of using a crossover rate of 0.5.

Name: Serban Bogdan Alexandru

ID: 1856482

- **Experiment 5: Average attendance vs population size**

In this experiment I have studied how increasing the size of the population can affect the average weekly attendance to the bar, and therefore the efficiency of the usage of the shared resource. The following parameter setting was used in this experiment:

- **Crossover ratio: r =** 200;
- **Mutation rate:** χ = 1.5;
- **Selection scheme:** Tournament Selection;
- **Tournament size: k** = 3;
- **Population size:** 50 ≤ **n** ≤ 1000.

For this experiment, I have decided to use population sizes ranging from 50 individuals, up to 1000, with an incrementing step of 50. The genetic algorithm was run 100 times for each individual value of the population size and the average weekly attendance at the bar was recorded. The results are plotted in the boxplots below.

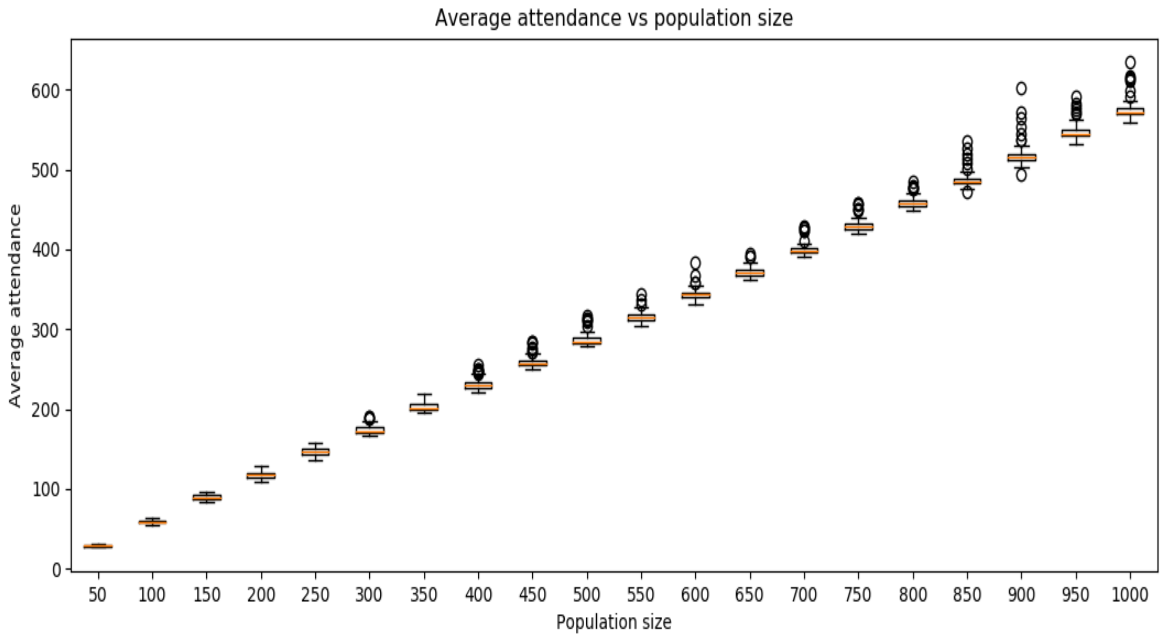

Figure 5 Average attendance vs population size

As it can be seen in **Figure 5,** there is a linear dependence between the average attendance to the bar (measured as the number of individuals) and the total population, the fact that was expected. To compensate for this aspect and to make the boxplots express the actual dependency between the population size and the performance of the algorithm, the attendance was turned into percentages out of the total population. The results are plotted below.

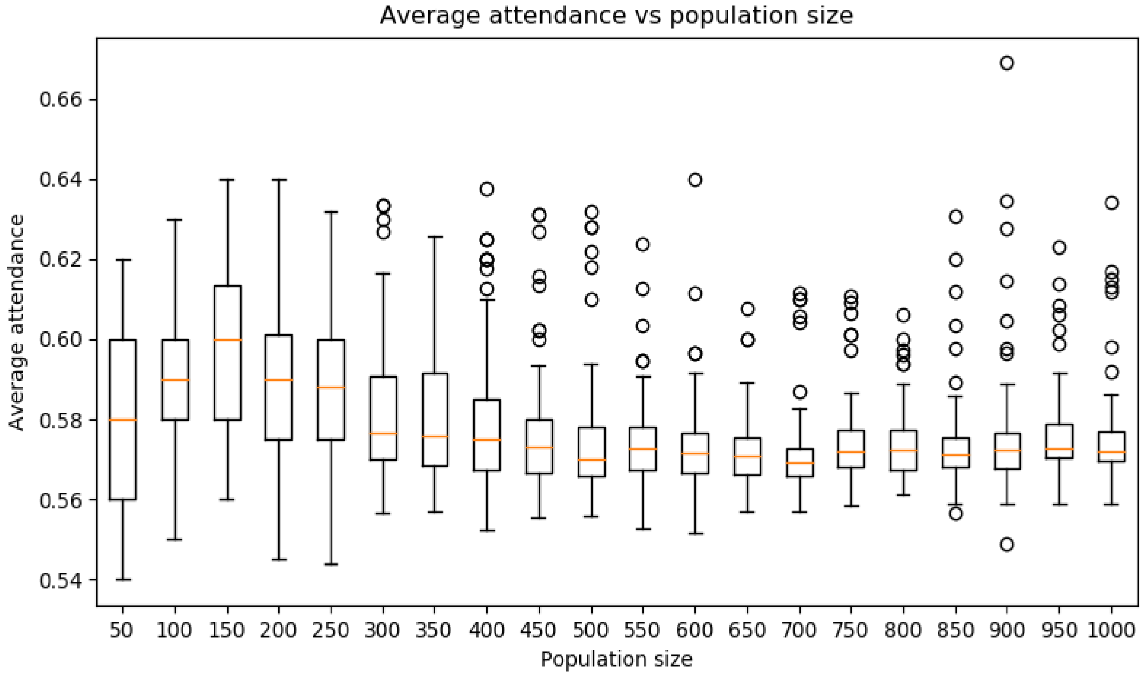

Figure 6 Average attendance (percentage) vs population size

As it can be seen in **Figure 6,** the size of the population does have a significant impact on the efficiency of the genetic algorithm, namely the attendance at the bar. One aspect that can be seen is the fact that having a population size that ranges between 50 and 450 individuals makes the algorithm to have good performance, with around 75% of the runs for each value of the population rate have given attendances of less than 60% of the total population size. One negative aspect of this is the fact that in 25% of the runs, the bar was overcrowded, meaning that the shared resource was over-utilized. Contrary to that, as the population size increases to 450 and gets up to 1000 individuals, the average attendance to the bar sets to 60% of the total population or below, with just a couple of extreme values that are outside of the interval.

Therefore, we can admit that for values of the population size of 450 or over, the algorithm will generate solutions that on average, will be able to use the shared resource at almost its maximum capacity, which means an efficient planning.