Finding the approach and the specifications for configuration

Kiarash Jamshidi

June 25, 2021

1 Introduction

Deepjanus is the tool for finding the frontier of the behavior. It has one branch for finding the behavior of the road with the road generating. It tests the lane-keeping assist Agent that it has. In this report, we add a new operator to DeepJanus, which is the Fog. At first, we investigated the way for generating the seeds and population. There were two ways of generating the data. The first way of generating RANDOM. RANDOM produces the Seed and initial population randomly from the fog range (in this report, the range is chosen between 0 and 0.4). The second way of generating is DIVERSITY which populates the Seed. Then the program test the Seed. If the program succeeds (the car did not go out of the lane and arrives at the last point), it would accept it and generate another seed. Otherwise, if the Seed during the test failed (it went out of the lane), it would create another road with another fog. The amount of the generating Seed is related to the amount of Poolsize in the configuration. In this report, the amount is equal to 20.

The first approach of generating Seed is RANDOM. In this report, we test all of the experiments with the same configurations. There are three aspects of the configuration that we want to find the best version of it in this program, which are 1-MutationFogPrecise, 2- MutationExtent 3- GeneratorName. GenaretorName is the approach for choosing how to generates the seed. After generating seed and initialing population. We should consider the meaning of the population in this approach. Every population is one Individual. There are two members in every Individual, the initial one and the mutant one. In the fog density, the initial member has a distinguish road shape and fog density. The mutation will change the fog density of the initial member with two factors 1- MUTATION-EXTENT, 2- MUTATION-FOG-PRECISE. Mutation-extend will choose randomly from the value that it receives. For example, if it gets 3, the value will be selected from between the range of (-3,3), which will be (-3, -2 -1, 1, 2, 3). the 0 is discarded because with zero will be no change. the whole formula is

$$Mutation = InitialValue + (MutationExtent * MutationFogPrecise)$$

The MUTATION-FOG-PRECISE is critical in the whole program. If this value decreases, the program accuracy will be increased. On the other hand, if the MUTATION-FOG-PRECISE has a higher value, the chance to find the success and failure member in one individual gets higher. So these two values should be chosen smartly.

The second approach for GeneratorName is the DIVERSITY approach. The DIVERSITY approach is the approach that works the same as the RANDOM, but at the beginning, it just chooses the successful seed. This means the beginning of the program test the road with fog density. If it fails, it changes the road and fog density, and if it was a success, it saves it and generates the next one.

First experiment has the configuration below:

```
MUTATION_TYPE = Config.MUT_FOG
FOG_DENSITY_threshold_max = 1
FOG_DENSITY_threshold_min = 0
FOG_DENSITY_threshold_for_generating_seed_max = 0.4
FOG_DENSITY_threshold_for_generating_seed_max = 0
MUTATION_FOG_PRECISE = 0.05
```

```
MUTATION_EXTENT = 6
generator_name = Config.GEN_RANDOM
POOLSIZE = 20
POPSIZE = 12
NUM_GENERATIONS = 10
ARCHIVE_THRESHOLD = 35.0
```

There are two thresholds, one for generating the seeds and the other threshold is for the mutation. Generating the seed range is between (0,0.3) and the mutation range is (0,1). The MUTATION-TY PE is the type of operation that the system has to run. In this experiment, we choose the fog operation, which the name in this program is MUT-FOG. Also, in this example, the approach is RANDOM, so generator-name = Config.GEN-RANDOM is for choosing the RANDOM approach as the first approach.

The pool size is the number of seeds which is 20, and Popsize is the number of individuals. For this test, Popsize is chosen 12, which means in the test area, we have 12 individuals who run the initial one member, and another member is the mutation one.

The number of generations is the amount of all four stages of the NSGA2 algorithm run once. For example, the 10 generation amount means the program will mutate the population, evaluate the population, update the archive, and select the fitness ten times.

The first section is comparing both approaches in duration of whole process per generation. The number of the archive will increase by every generation. The pace of this duration growing is essential for our program. The fourth section is the standard deviation of each generation. If the standard deviation is getting a high amount, the program and the frontier have better quality.

2 Duration of process

In this part, we want to see how much time every generation takes..

2.1 RANDOM

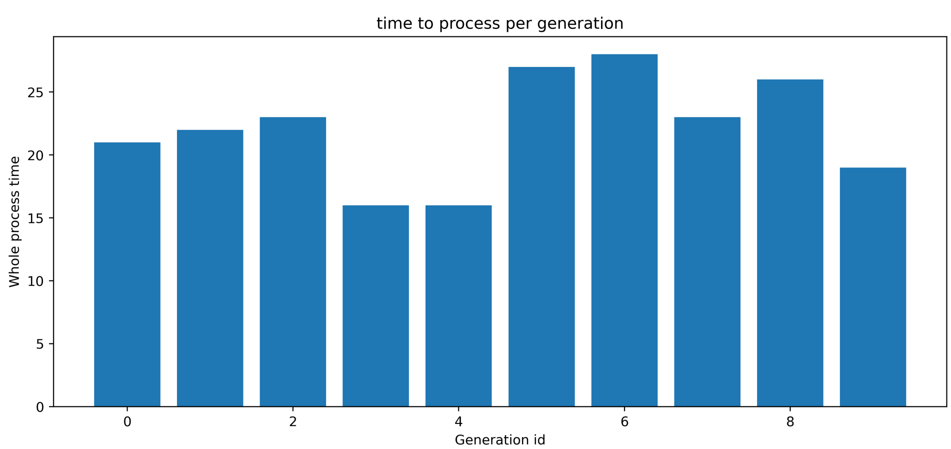

Figure 1: Whole process duration per generation with MUTATION-FOG-PRECISE=0.05 and MUTATION-EXTENT=6

As you can see in Figure 1, the time for every generation is between 20 and 25 minutes. In the second configuration, we change the two values to see there is any difference or not.

In the new this configuration is only changed by MUTATION-FOG-PRECISE, In the last example, the value was 0.05, and we decrease it to 0.01. Furthermore, all the rest of the configuration is the same as the first experiment.

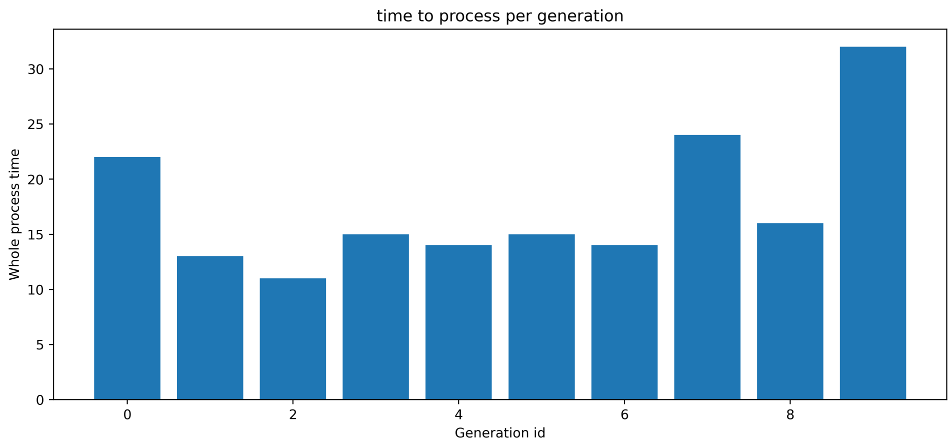

Figure 2: Whole process duration per generation with Mutation-Fog-Precise=0.01 and Mutation-Extent=6

By comparison these two experiments, we noticed that the average time is pretty the same for both of them. However, for the fewer Mutation-Fog-Precis, the duration at the beginning was less than the other experiment. Also, we noticed that the average time duration is pretty the same for both of them. However, for the less Mutation-Fog-Precis, the duration at the beginning was less than the other experiment.

2.2 DIVERSITY

In the previous part, we saw the experiment by changing the MUTATION-FOG-PRECISE. The first experiment is with MUTATION-FOG-PRECISE=0.02 and MUTATION-EXTENT=6. Figure 3 is the result.

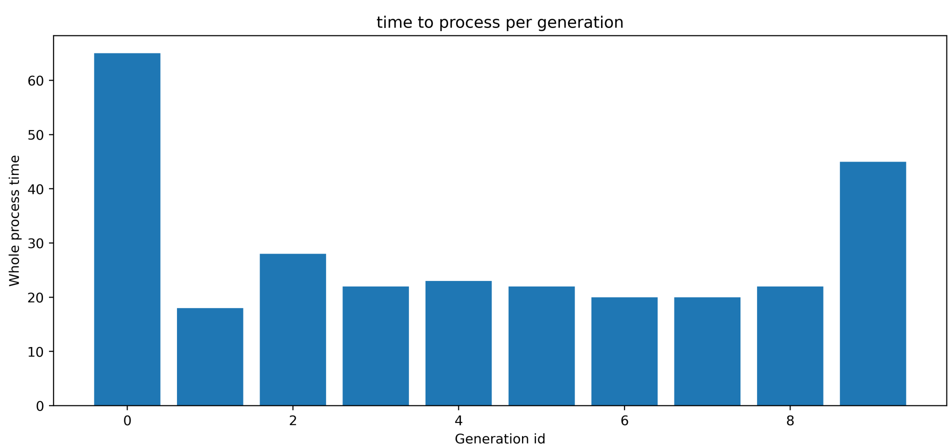

Figure 3: Whole process duration per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.02, and MUTATION-EXTENT=6

Compared with the RANDOM approach, it has a big difference in the first generation. in the second experiment, we change the MUTATION-FOG-PRECISE to see it. we decreased it from 0.02 to 0.01 and the Figure 4 is result.

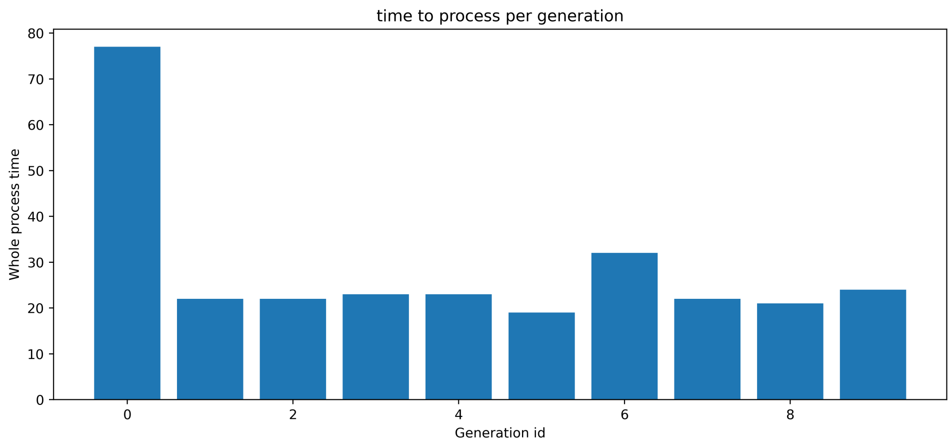

Figure 4: Whole process duration per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

We did not find the difference between this two, maybe it decreased a bit, but it was not noticeable. So we went for the second option, which is MUTATION-EXTENT. In the third experiment, this amount change from 6 to 4, and Figure 5 is showing the outcome.

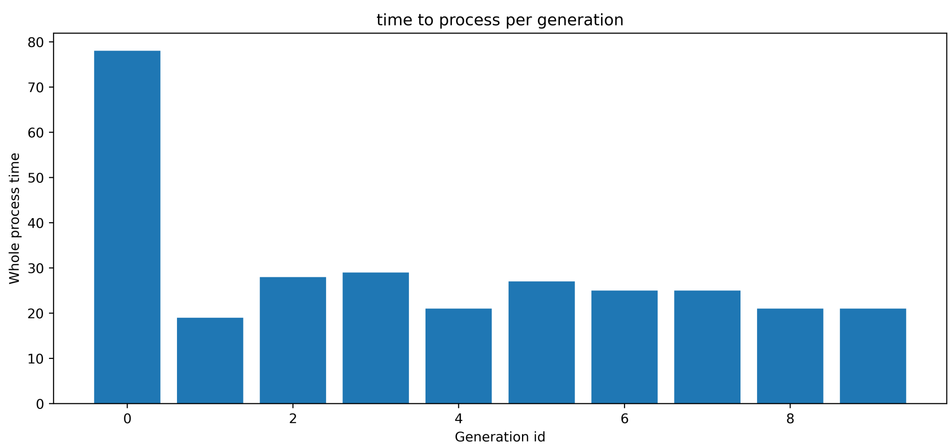

Figure 5: Whole process duration per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

Also, the modification is not noticeable in this version, so the two factors of MUTATION-FOG-PRECIS and MUTATION-EXTENT have no effect or low effect on the duration.

2.3 First generation differences

In the experiment, we understand a difference between the timing of the first generation in the RANDOM and DIVERSITY version, so we divided the whole process time slot into three distinct parts. The first part is initialing the population, the second part is evaluating the population by testing them, and the third part is selecting the archive and select the fitness.

The first attempt is for the RANDOM approach, which with MUTATION-FOG-PRECISE=0.01 and MUTATION-EXTENT=6 .the result is in Figure 9. The blue part belongs to initialing population, the green part belongs to the evaluation of population, and the red is part for selecting archive and fitness.

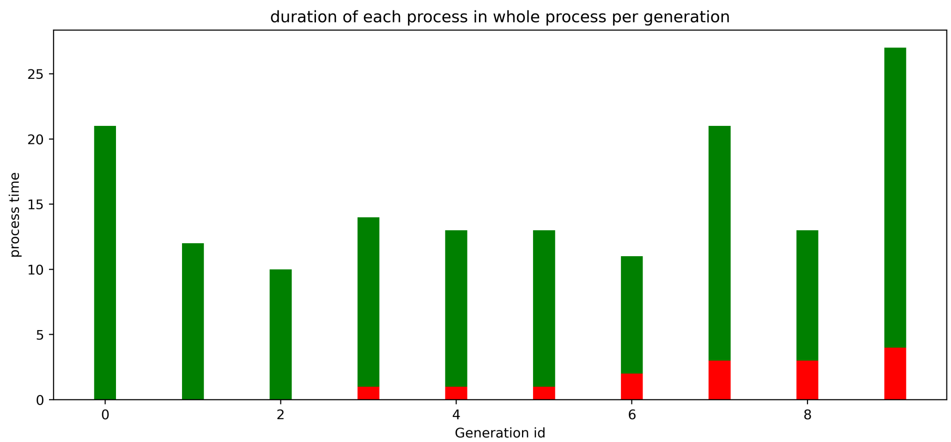

Figure 6: Different time slots per generation with RANDOM-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

The Figure shows us that the initialing population in the RANDOM approach takes a little time. Also, most of the time in every generation belongs to the evaluation of the population. Furthermore, selecting the archive and selecting the fitness part is growing by generation.

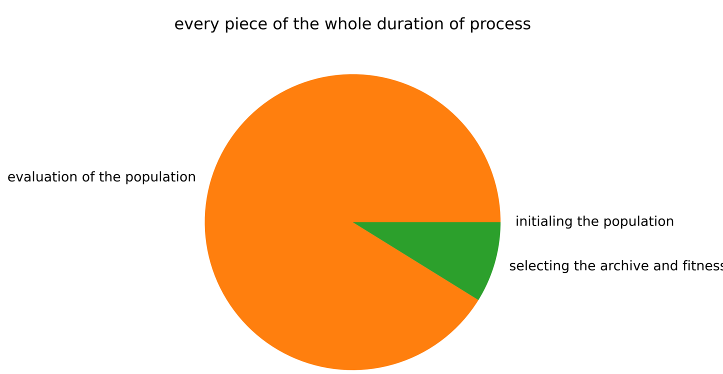

Figure 7: Different time slots for whole program(all generations together)with RANDOM-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

The second experiment belongs to the DIVERSITY approach. With the same configuration, Figure 8 is produced.

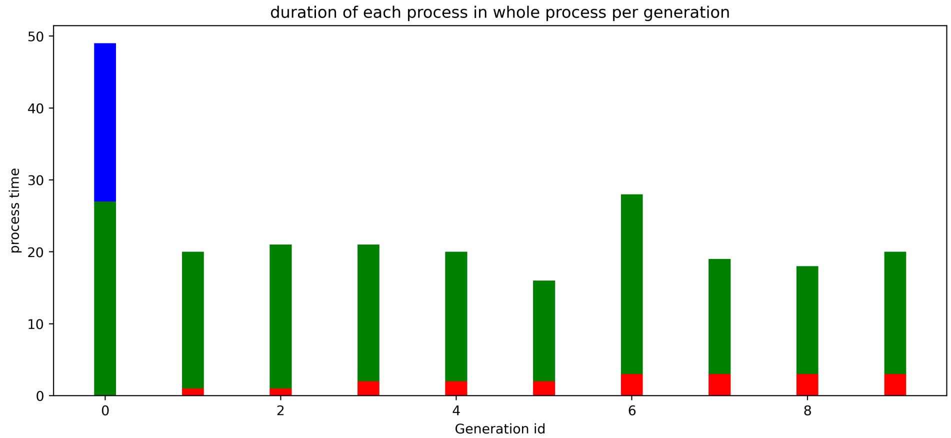

Figure 8: Different time slots per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

The only concern about the time slot is increasing the time slot to select the archive and select the fitness. If you look at both bar charts, the red bar is growing dramatically for both approaches. This growth will become a problem for the testing with 100 generations. On the other hand, we should look at the size of the archive every approach will take. In the next section, it shows the number of the archive produced by program per generation.

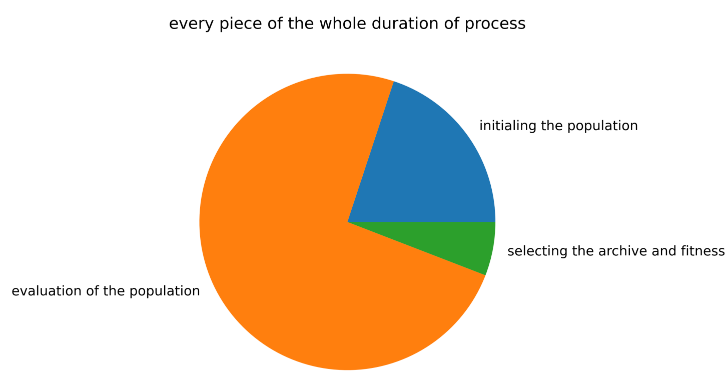

Figure 9: Different time slots for whole program(all generations together) with RANDOM-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

So overall, the initialing population is a big part of the whole process duration. But it is fix number. So for test cases with a higher number of generations, this initialing population duration won't play a central role in the total period of the process.

But the one that has more concerning is the selecting archive and selecting fitness part (red in the bar chart). It is growing because every process number of the archive is rising, so the time for calculation spareness fitness will increase. So we should take care of the size of the archive to avoid wasting time. The next part compares the size of the archive with different amounts of three values (Mutation-Extent, Mutation-Fog-Precise, Generator-Name).

3 Size of the archive per generation

Each generation has its archive size. The archive size means the number of the individual which has pair of the failure and successful members. in this section. We look at the two different effects of the values on the archive producing size per generation.

3.1 RANDOM

Like the last version I test the RANDOM approach first, as we know we go for the first configuration and with Mutation-Fog-Precise=0.05 and Mutation-Extent=6

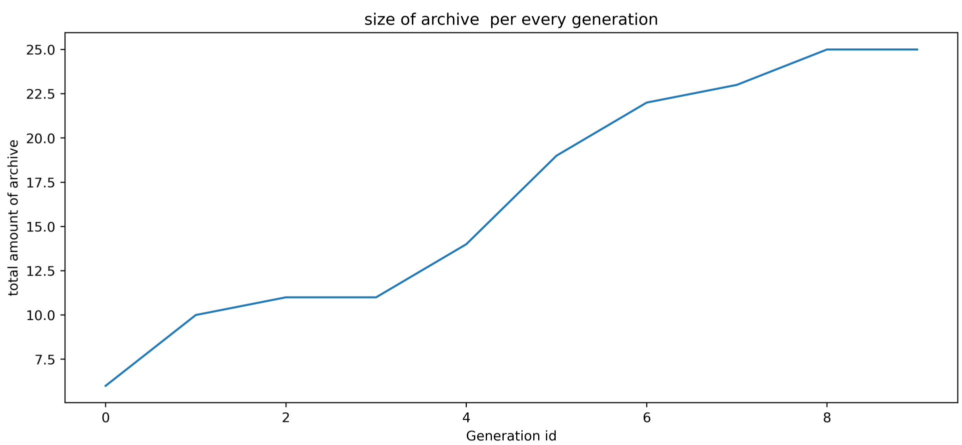

Figure 10: Size of Archive per generation with RANDOM-generator, MUTATION-FOG-PRECISE=0.05, and MUTATION-EXTENT=6

In the second test, the amount of the Mutation-Fog-Precise was modified, it decreases from 0.05 to 0.01, and we want to see how the system reacts to it.

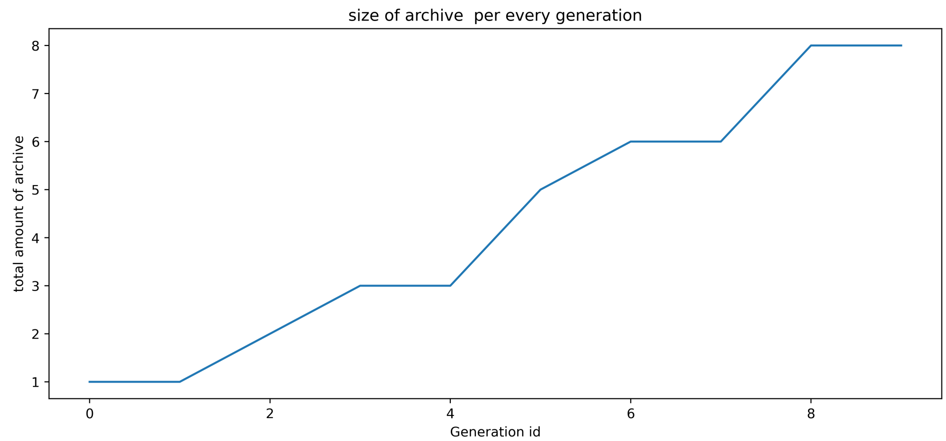

Figure 11: Size of Archive per generation with RANDOM-generator, MUTATION-FOG-PRECISE=0.01 and, MUTATION-EXTENT=6

As we can see, the higher amount of the with Mutation-Fog-Precise produces more individuals for the archive, which means archive size is getting bigger. On the other hand, decreasing the value of Mutation-Fog-Precise will reduce the size of the archive dramatically from 25 to 6.

3.2 Diversity

By testing RANDOM, we assume decreasing the amount of MUTATION-FOG-PRECISE will affect the size of the archive, and it will be the cause of decreasing. In this first experiment of the DIVERSITY approach, we test the program with Mutation-Fog-Precise=0.02 and Mutation-Extent=6. Figure 16 is showing the outcome.

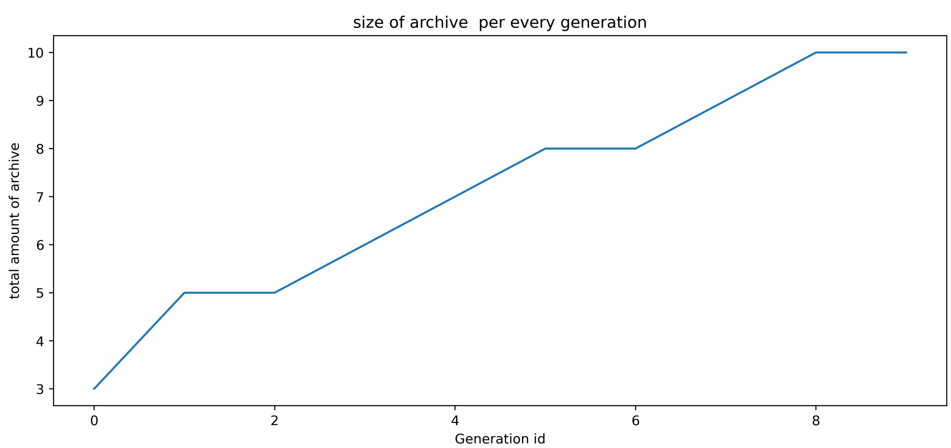

Figure 12: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.02, and MUTATION-EXTENT=6

In the second test, the amount of the Mutation-Fog-Precise was modified, it decreases from 0.02 to 0.015, and the result is in Figure 13

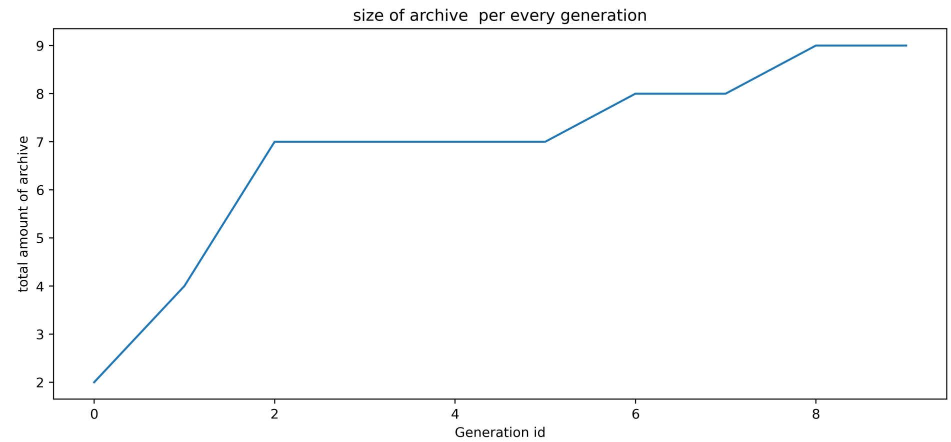

Figure 13: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.015 and, MUTATION-EXTENT=6

The change of the last two experiments was only one. The number dropped from 10 to 9, so we decrease the number from 0.015 to 0,01 to see how the system reacts to it. The result is in the Figure 16.

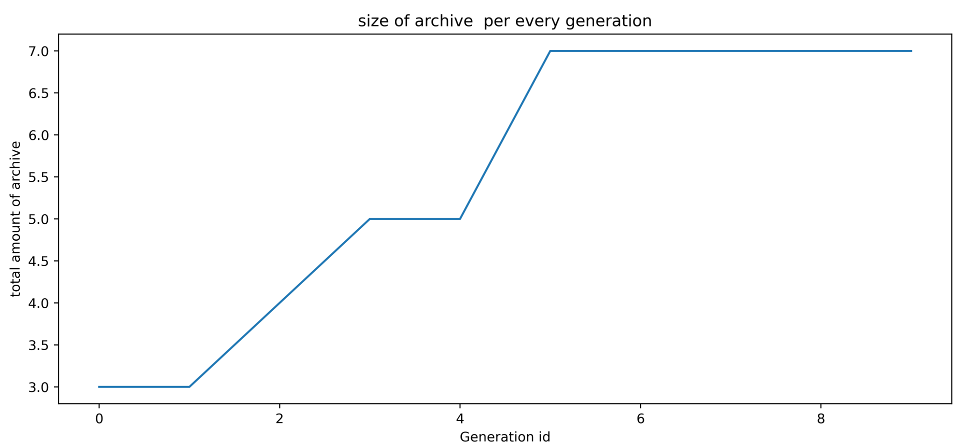

Figure 14: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6

So, in conclusion, we can see the relation between the number of archives and MUTATION-FOG-PRECISE. However, there is another factor that is essential for us. MUTATION-EXTENT is another factor that can affect the experiment. So in the first test, we did not change the number of MUTATION-FOG-PRECISE. We decreased the number of MUTATION-EXTENT from 6 to 4. and Figure **??** is the result of our experiment.

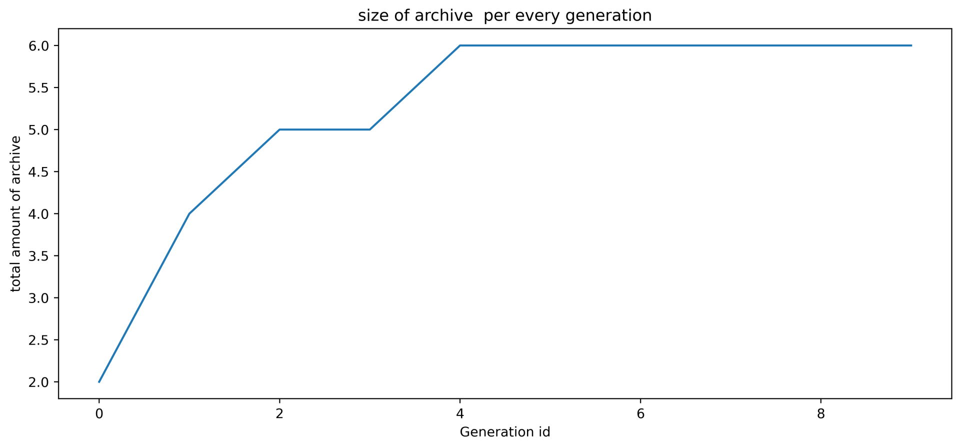

Figure 15: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=4

As we can see, it has the same result as MUTATION-FOG-PRECISE. The number decreases from 7 to 6. so also this feature is essential.MUTATION-EXTENT but may cause some abnormality that you can see in the **??**.

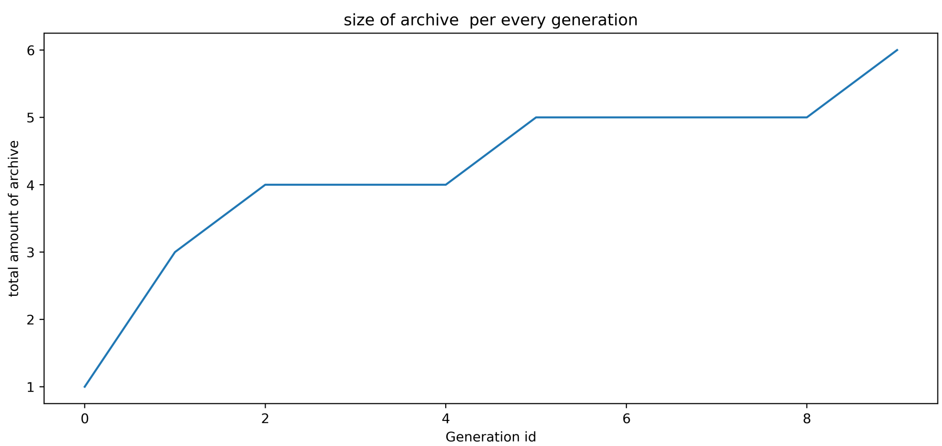

Figure 16: Size of Archive per generation with MUTATION-FOG-PRECISE=0.01 and MUTATION-EXTENT=8 , DIVERSITY-generator

The number was decreasing when we raised it from MUTATION-EXTENT = 6 to MUTATION-EXTENT = 8. The size of the archive decreased from 7 to 6. Also, the new question will be how we can choose the best configuration for the frontier. In this step, we encounter the concept of standard deviation, which we will find out in the next step.

4 standard deviation

The standard deviation plays a substantial role in every testing. If we have a better standard deviation, it would be a better test. the standard deviation shows the difference in amount between the frontier, which we find for the road. As usual, we go for the Random approach first.

4.1 RANDOM

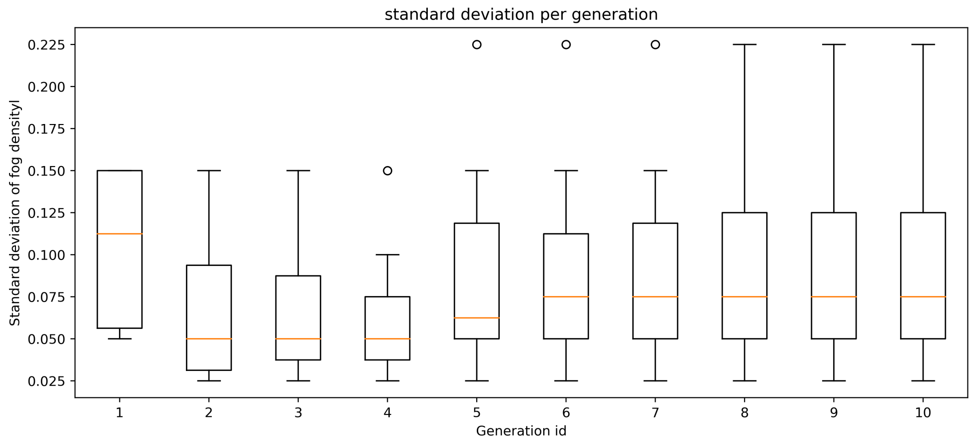

Figure 17: Standard deviation of fog density average per generation with RANDOM-generator,MUTATION-FOG-PRECISE=0.05 ,and MUTATION-EXTENT=6

we evaluate the relation by the MUTATION-FOG-PRECISE by decreasing it 0.01. the new configuration will be MUTATION-FOG-PRECISE=0.01 ,and MUTATION-EXTENT=6.

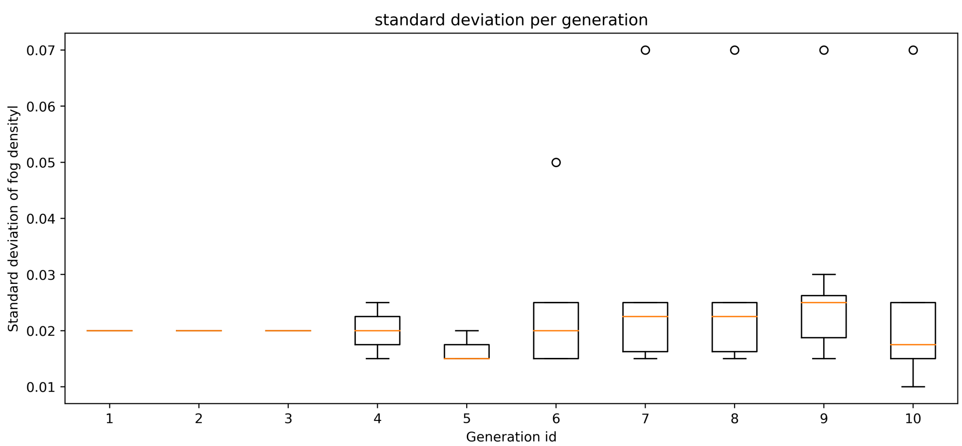

Figure 18: Standard deviation of fog density average per generation with RANDOM-generator,MUTATION-FOG-PRECISE=0.01 ,and MUTATION-EXTENT=6

The number increases from a range of (0, 0.055) to a range of (0, 0.07) it may happen accidentally. We also test it in the RANDOM approach.

4.2 DIVERSITY

IN the first experiment, as always. It will start with configuration MUTATION-FOG-PRECISE=0.02 and MUTATION-EXTENT=6.

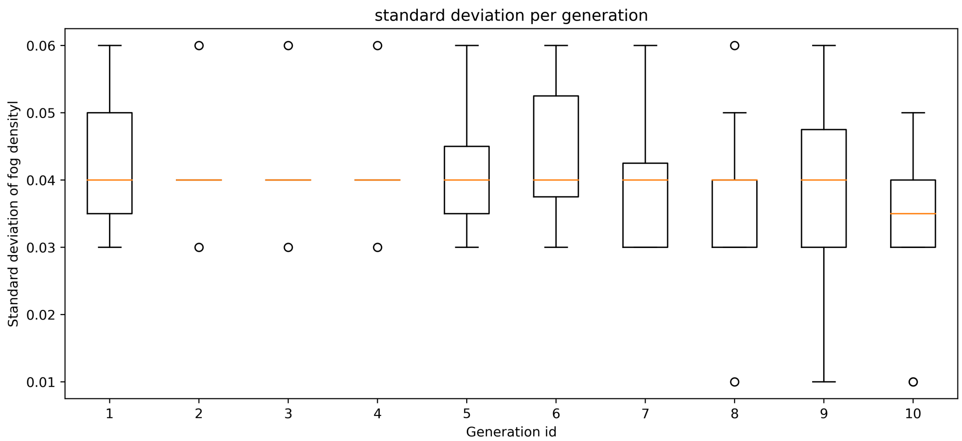

Figure 19: Standard deviation of fog density average per generation with DIVERSITY-generator,MUTATION-FOG-PRECISE=0.02 ,and MUTATION-EXTENT=6

Then we assess the MUTATION-FOG-PRECISE by decreasing it to 0.01. the new configuration will be MUTATION-FOG-PRECISE=0.01 and MUTATION-EXTENT=6.

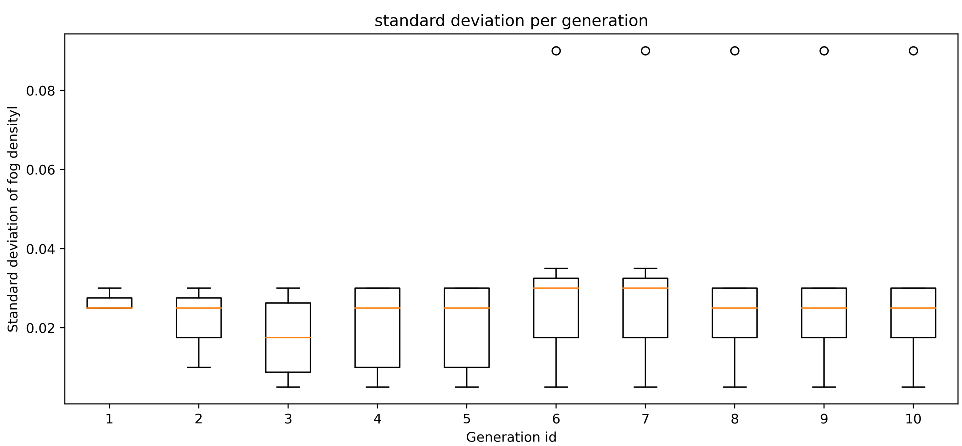

Figure 20: Standard deviation of fog density average per generation with DIVERSITY-generator,MUTATION-FOG-PRECISE=0.01 ,and MUTATION-EXTENT=6

By comparing the two trend, you will notice the modifications of MUTATION-FOG-PRECISE was not effective. The RANDOM will ma change accidentally, but here it is proven that the number is not related.

Now, it is time for the changing the second factor. MUTATION-EXTENT is decreasing by 2. From 6 into the 4. Figure 21 is showing the trend.

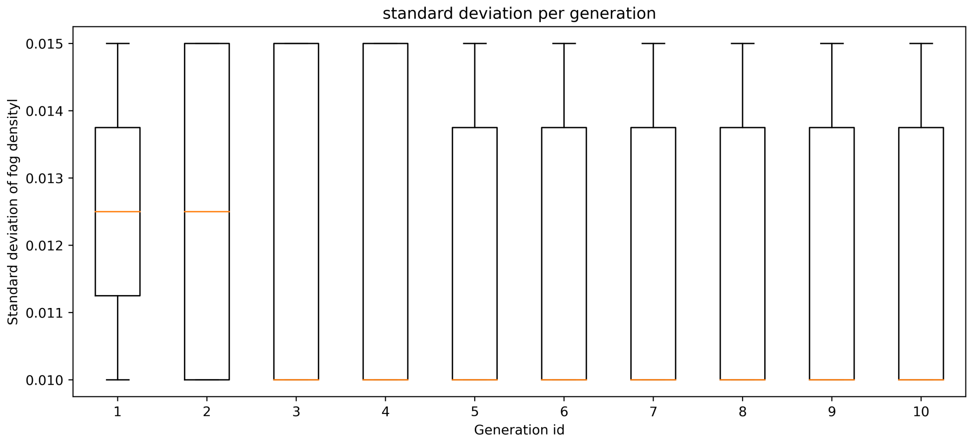

Figure 21: Standard deviation of fog density average per generation with DIVERSITY-generator,MUTATION-FOG-PRECISE=0.01 ,and MUTATION-EXTENT=4

This experiment is produced by increasing the number of MUTATION-EXTENT from 6 to 8. Figure 22 is showing the trend.

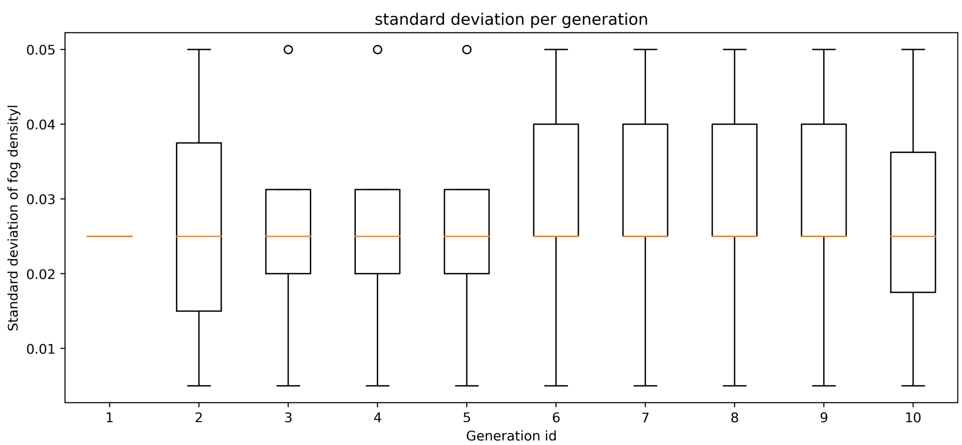

Figure 22: Standard deviation of fog density average per generation with DIVERSITY-generator,MUTATION-FOG-PRECISE=0.01 ,and MUTATION-EXTENT=8

The value of the standard deviation is increasing by increasing the number MUTATION-EXTENT from 4 to 8 by 0.005(from range of (0, 0.030) to range of (0, 0.035)).

On the other hand, the standard deviation is decreasing by increasing the number MUTATION-EXTENT from 6 to 8 by 0.35 (from range of (0, 0.70) to range of (0, 0.035)).

So, we can find any relation between the standard deviation of fog density and MUTATION-EXTENT.

5 Conclusion

This report tries to understand the best solution for the fog density specifications in the configuration and the best approach.

The first conclusion is the approach of the generation. According to section number two, the time difference between these two approaches is not noticeable. Only the DIVERSITY version tests the seed firstly, which takes like 30 to 40 minutes more time. Also, both ways have the same pace of increase in the size of the archive per generation with the same configuration. But the noticeable point was the size of the archive, which is too high for RANDOM. IN the time duration for selecting the archive and fitness (spareness calculation). We noticed with the larger size of the archive will be more time for this part. Usually, the test for the program is more than 100 generations, so with the RANDOM approach, the size will be super big, and the duration of spareness calculation will be huge too. And also, the standard deviation seems to have the better result in the DIVERSITY approach. So, as a result, the DIVERSITY seems more reasonable than RANDOM for our test.

Also, for EXTENT-MUTATION, we understand the value of 4 was too low for the size of the archive(it makes a small amount of archive). Also, the value of 8 is too high, and it will encounter with big size of the archive. So the best result was 6. And the last variable, which was MUTATION-FOG-PRECISE, the best result was 0.01 according to different tests that we had.So as a conclusion, this configuration will be best for our operator:

```
MUTATION_FOG_PRECISE = 0.01
MUTATION_EXTENT = 6
generator_name = Config.GEN_RANDOM
```

We also can find these three variables for the new operator in the same way.