Finding the best form for selecting archive

Kiarash Jamshidi

June 25, 2021

1 Introduction

Selecting the archive is one of the critical steps in the Deepjanus. There are different strategies and algorithms for choosing the individual in the archive. Also, there are a lot of aspects that have a significant influence on this step. In the Deepjanus, there are already two types of selecting archives already implemented in the tool. For the new operation. On the Deepjanus, we try to find out which one has better results. In this report, I explain the two ways of selecting the archive, then looking forward to assessing the different aspects of the program. And the last part, the conclusion is explaining the result of the comparison. In another way, we decide to see which approach is better for the new operator.

The two experiments in the report were taken with the same configuration. The configuration is shown 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_min = 0
MUTATION_FOG_PRECISE = 0.01
MUTATION_EXTENT = 6
generator_name = Config.GEN_DIVERSITY
POOLSIZE = 20
POPSIZE = 12
NUM_GENERATIONS = 10
ARCHIVE_THRESHOLD = 35.0
```

2 Greedy archive

The first way is the greedy archive. In the program, every individual has two members. The initial one always will be successful because, in the configuration, we choose the generator-name = Config.GEN-DIVERSITY, which means the seed of the population is successful (successful means the member has the specification which the program test the lane-keeping assist, It doesn't go out of the lane reaches the last point of the road). So the mutant one will be the failure one. The greedy approaches save all the individuals with this feature (which means one member cause of the success and the other member (mutant one) cause of the failure). The Greedy-Archive captures all of the individuals with those conditions.

2.1 Size of archive per generation

The population will grow per generation. have more archives is better for the investigation. In this part, we show the size of the total archive per generation. In every generation 12 individuals will be tested. Every individual has pair of members. If the pair of members have different behavior (one is successful and the other member is failure), the program will save it in the archive. So in every generation, the number of the archive will grow. The trend shows the total size of the archive per generation. The result is in Figure 1.

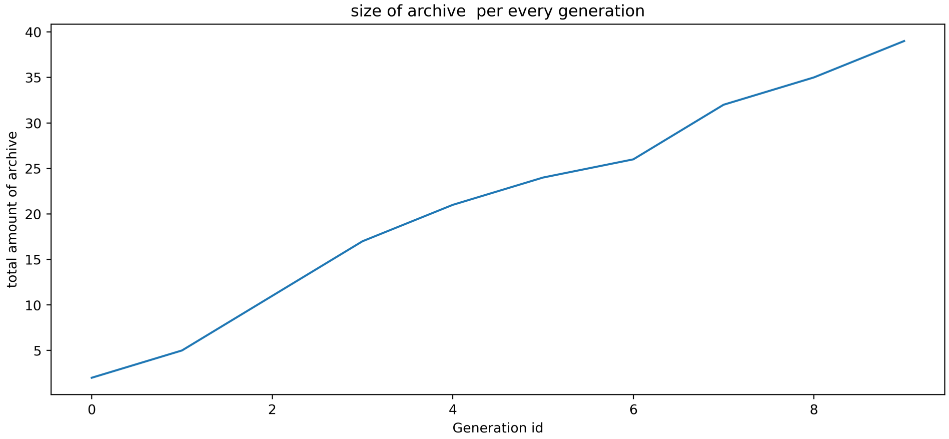

Figure 1: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

At the 10th generation, we have approximately 40 individuals in the archive.

2.2 Duration of process per population

Time of the process is also another important factor. In this part, we want to see how much time every generation takes.

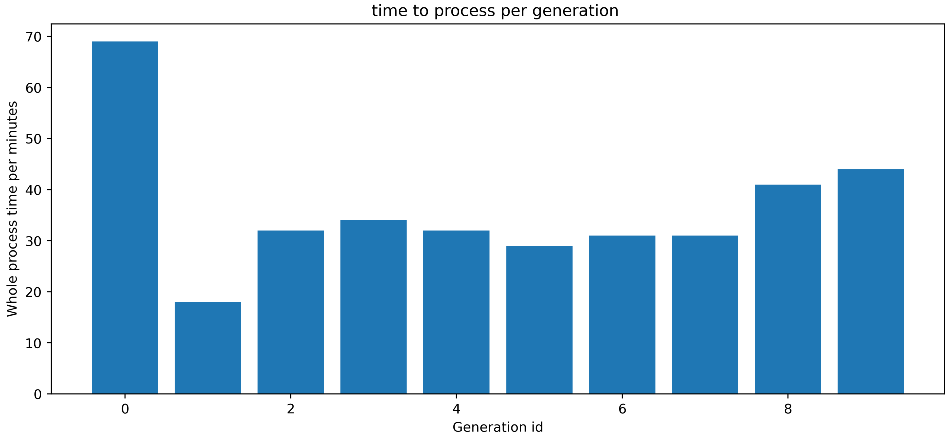

Figure 2: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

The first generation has a lot more time than the others. And it raises per generation. In the next part, we look at the duration in more detail to understand the different aspects of this criteria.

2.2.1 Whole process duration in pieces

We divided the whole time f process into three pieces. The first one generates the population, which generates the seed and the population for the process(the color is blue in the bar chart). The second part is the duration of evaluating the process, which is testing every individual pair to see which one will fail and which one will succeed(the color is green in the bar chart). The third part is selecting the archive and the fitness, calculating all the distance and choosing the archive, etc. (the color is red in the bar chart). Figure 3 is showing the result.

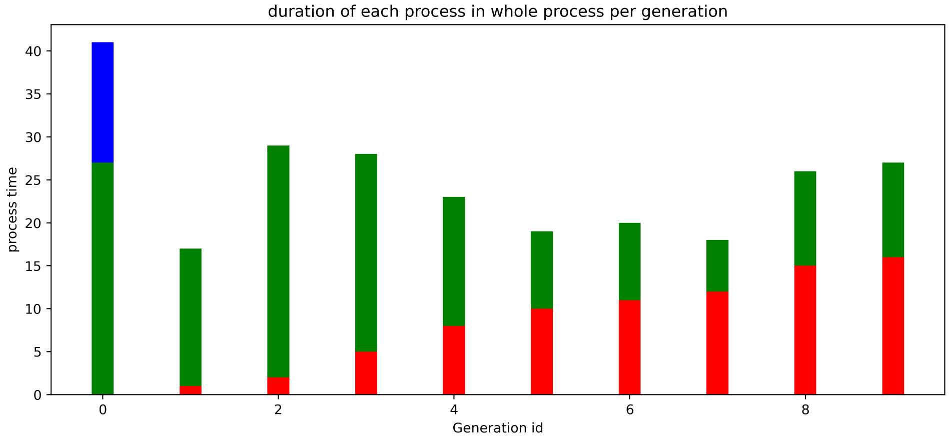

Figure 3: each piece of process per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

The first generation has a lot of time for the "generating the population" step. It happens because the program considers the generating seed as generating population for the first generation. But as you can see in the other generation, the first part is not considered the long duration part. In the pie chart 4, we show the amount of every part in the whole time of the process.

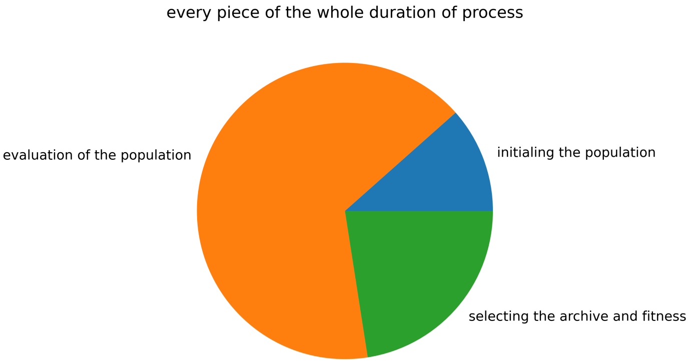

Figure 4: Whole time of the process with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

2.2.2 Selecting archive and selecting fitness duration per generation

In the last part, we notice raising the duration of the "selecting archive and selecting fitness" region. So in this section, Figure 5, is showing the pace of the growing the duration per generation.

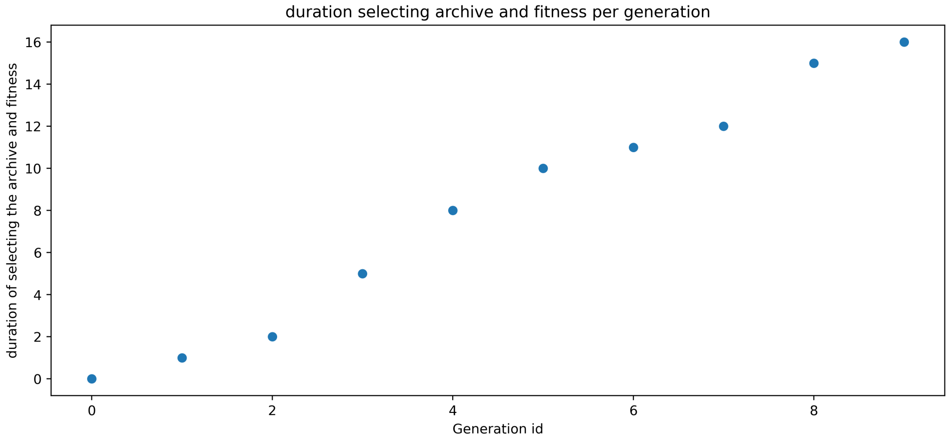

Figure 5: selecting Archive in minutes per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

The time is growing dramatically per generation.

2.3 Standard deviation of fog density per generation

In this process, the most critical fact is calculating the fog density and finding the best frontier for it. The standard deviation of the average fog density per generation is shown in Figure 6. .

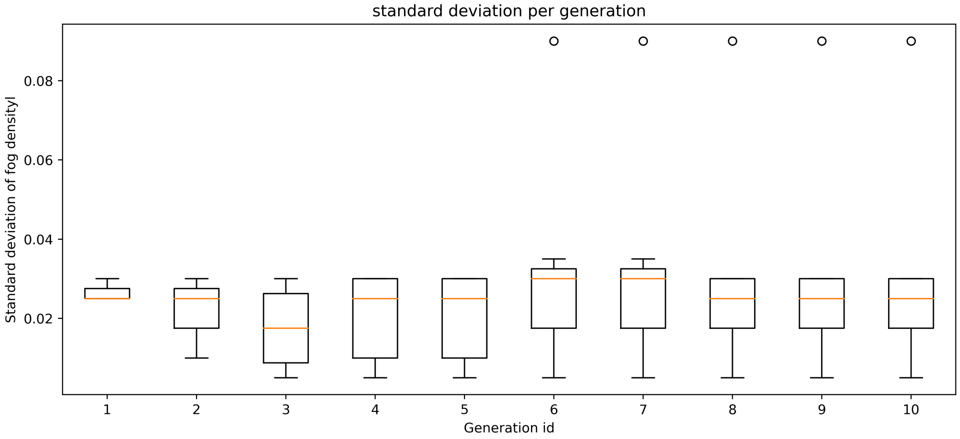

Figure 6: standard deviation per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6 , Greedy archive

3 Smart archive

The second way is the Smart archive. In contrast to Greedy-Archive, Smart-Archive has some conditions for collecting the archive. At first, it takes all individuals in the archive. Then it defines one threshold for the archive. In this experiment, the ARCHIVE-THRESHOLD is equal to 35. the program has two distances in the fog operator. The shape of the road calculates the first distance, and the difference in the fog density calculates the second distance. Every individual may keep it in the archive if it has the first condition (one member fails and the other success). If the distance between the new pair and the ones in the individual is too large (over 35), it means the individual is far from any individuals. So the individual is not suitable for us so that we won't save it. In the second condition, we compare the candidate with each other. The candidate with less distance between the members of each pair(the closer member) will be saved. Let see how this method will work with our configuration.

3.1 Size of archive per generation

Same as the first experiment, we compare it ion the size of the archive in the generation.

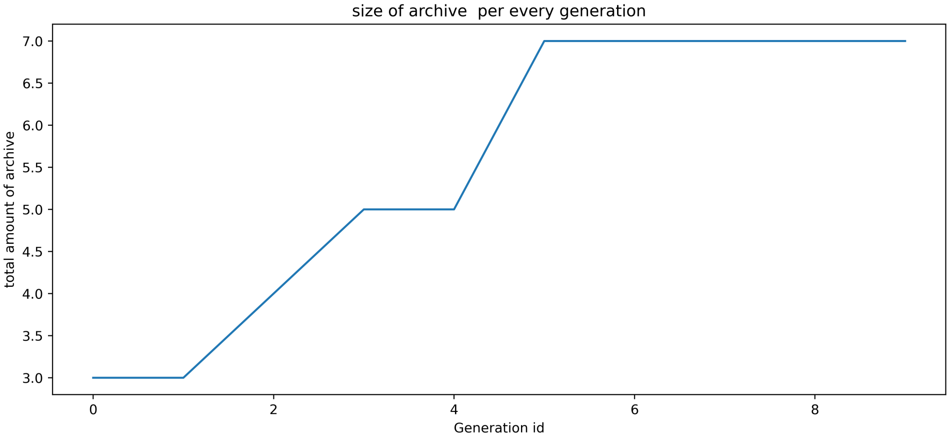

Figure 7: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

The big difference between the Smart archive and the Greedy archive is the size of the archive. It decreases from approximately 40 in Greedy-Archive to 7 in Smart-Archive. In the next section, we will look at the time duration of the process.

3.2 Duration of process per population

The time-consuming process for each generation is shown in Figure 8.

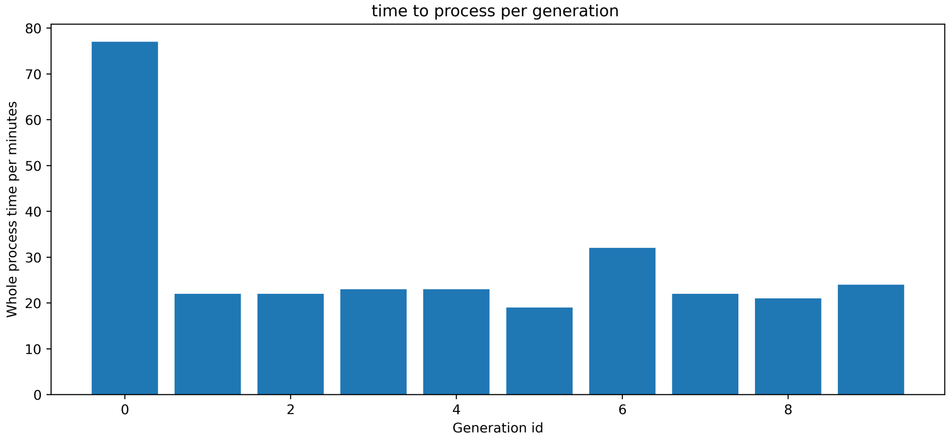

Figure 8: Size of Archive per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

AS you can see, this duration is less than the previous one in the last generations. Now, let see the time in more detail.

3.2.1 Whole process duration in pieces

Again, we introduce the decomposition of the whole process time to better look at the details. We separate it into 3 part which is: 1-initialing the population "blue", 2- evaluation of the population "green", 3 - the selecting the archive and selecting the fitness "red".

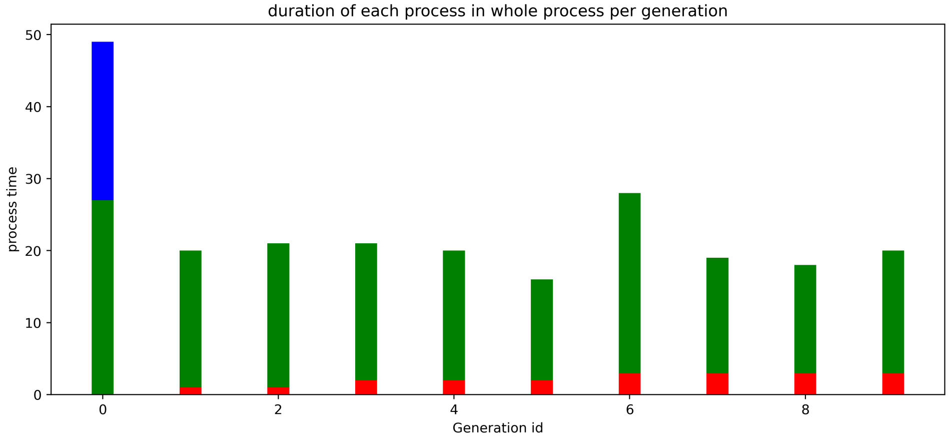

Figure 9: each piece of process per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

As we can see, the amount of the third step decreases a lot which means selecting the archive and fitness now has less time duration than before. The first two parts (initialing the population and evaluating the individuals are similar) in both approaches.

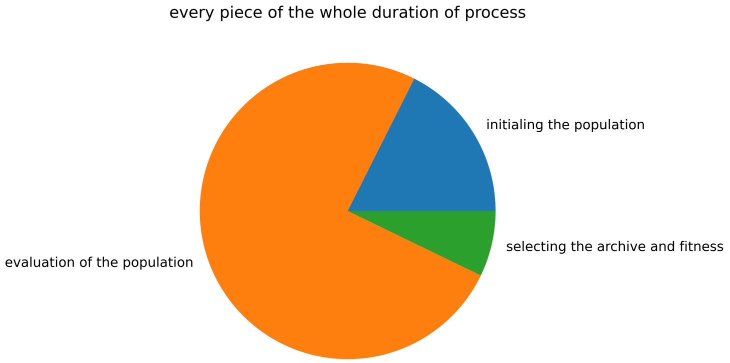

Figure 10: Whole time of the process with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

Also, the third part is decreasing in the whole amount of the process.

3.2.2 Selecting archive and selecting fitness duration per generation

We understand from the last part that selecting the archive and fitness takes less time than the first approach. Therefore, in this part, we want to see the size of the duration per generation and compare both methods' growing speed.

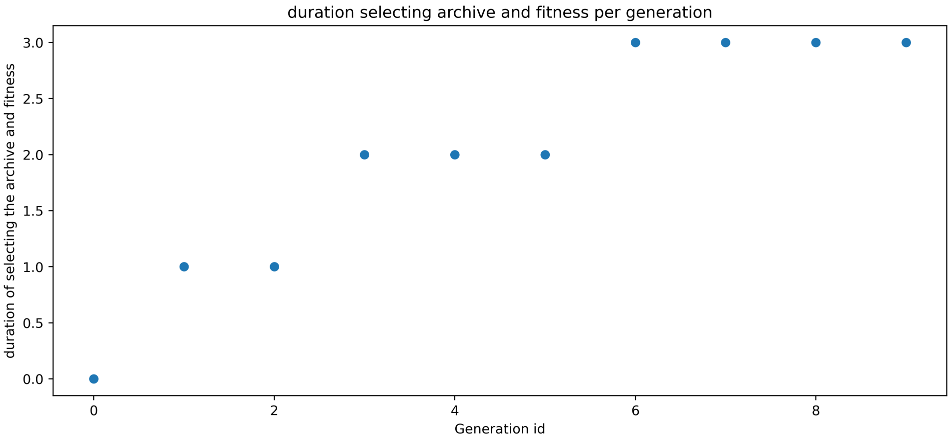

Figure 11: selecting Archive in minutes per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

Obviously, with the Smart-Archive, we have less growing speed.

3.3 Standard deviation

The last part belongs to the standard deviation of the fog density per generation. Figure 12 is showing the result.

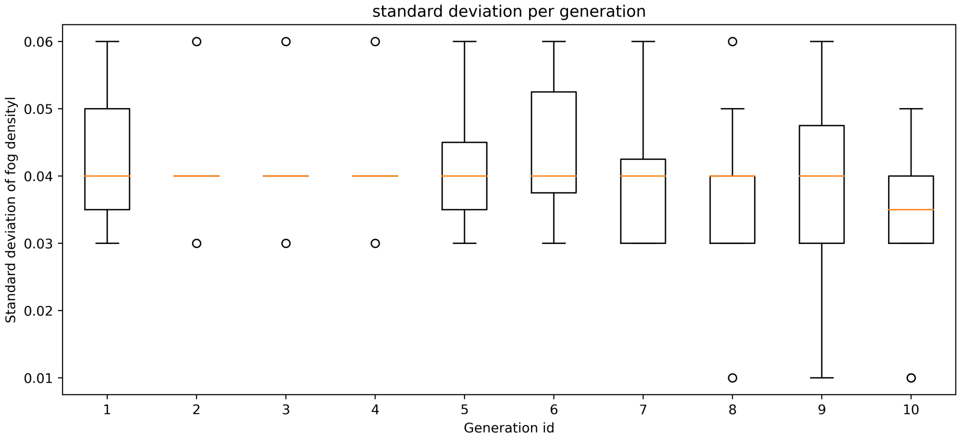

Figure 12: standard deviation per generation with DIVERSITY-generator, MUTATION-FOG-PRECISE=0.01, and MUTATION-EXTENT=6, Smart archive

In this approach, we have a lower standard deviation.

4 Conclusion

One of the essential factors in our process is time. Although the Greedy-archive provides us more archives, it takes too much time for the further generation. Also, in this experiment, the generation was 10, but in the actual test, the size of the generation will be over 100, which will be a lot of time for the Greedy-approach. The other fact is the standard deviation of the Smart-Archive was better. So this report shows for the new operator, the Smart-archive will be a more intelligent choice than the Greedy-archive.