

Assignment 2.

Metaheuristics



José Manuel Izquierdo Ramírez

Contents

1	Exercise 1.	3
1.1	Choosing a complex problem.	3
1.2	Analyzing the behavior when we modify the hyperparameters.	4
1.3	Conclusions of the behavior of the algorithm	8
1.4	Do you always get the best solution? Why? What does it depend on?	9
1.5	Modify the code to incorporate elitism. The best solution is kept in the elite and it is never lost until a new better one is obtained. This solution is the one that is returned at the end. Have you managed to improve? Why?	10
1.6	Conclusions of the behavior of the algorithm with elitism	14
1.7	So far, we have started with valid solutions. Change it so that any solution can be generated. Does it affect the final performance? Analyze this issue.	15
1.8	Conclusions of the behavior of the algorithm with elitism	19

List of Figures

1	Accuracy when the number of items increase	3
2	Statistics when we change the number of solutions	4
3	Statistics when we change the maximum number of generations	5
4	Statistics when we change the tournament size	6
5	Statistics when we change the probability of crossover	7
6	Statistics when we change the probability of mutation	8
7	Statistics when we change the number of solutions and use elitism	10
8	Statistics when we change the maximum number of generations and use elitism .	11
9	Statistics when we change the tournament size and use elitism	12
10	Statistics when we change the probability of crossover and use elitism	13
11	Statistics when we change the probability of mutation and use elitism	14
12	Statistics when we change the number of solutions and start with invalid solutions	15
13	Statistics when we change the maximum number of generations and start with invalid solutions	16
14	Statistics when we change the tournament size and start with invalid solutions .	17
15	Statistics when we change the probability and start with invalid solutions	18
16	Statistics when we change the probability of mutation and start with invalid solutions	19

1 Exercise 1.

How does the algorithm behave as we modify the number of solutions, generations, tournament size, crossover probability, and mutation probability? Add more objects to the problem to check the behavior of the algorithm.

1.1 Choosing a complex problem.

To perform these measurements we will first start by evaluating the behaviour of the algorithm when we add more objects. We add more objects in order to evaluate the behaviour of the algorithm when solving a sufficiently complex problem.

We will start by evaluating 100 times each entry of the dataset reflected in the input.txt file, which contains 5,6,7,8,10,15 and 24 items.

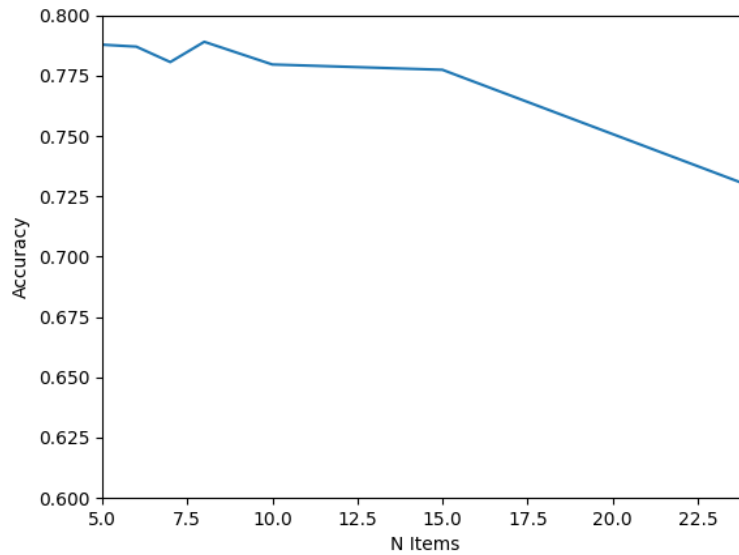


Figure 1: Accuracy when the number of items increase

A visual analysis shows that the accuracy decreases as the number of items increases, Pearson's correlation coefficient corroborates this, as the relationship between both variables is -0.93 with a certainty of 99.7%.

Next we will evaluate the behaviour of the metaheuristic when we modify each of the hyper-parameters, these will be evaluated for the following problem:

- Max Weight = 6404180
- Weights = [382745,799601,909247,729069,467902, 44328, 34610,698150,823460, 903959,853665,551830,610856,670702,488960,951111,323046,446298,931161,31385,496951, 264724,224916,169684]

- Prices = [825594, 1677009, 1676628, 1523970, 943972, 97426, 69666, 1296457, 1679693, 1902996, 1844992, 1049289, 1252836, 1319836, 953277, 2067538, 675367, 853655, 1826027, 65731, 901489, 577243, 466257, 369261]

1.2 Analyzing the behavior when we modify the hyperparameters.

To analyse the behaviour we will carry out 100 iterations for each of the values of each hyperparameter. We will analyse the accuracy, mean, variance and standard deviation of the knapsack value and the Pearson's correlation coefficient between accuracy and the hyperparameter in particular.

Modifying the number of solutions

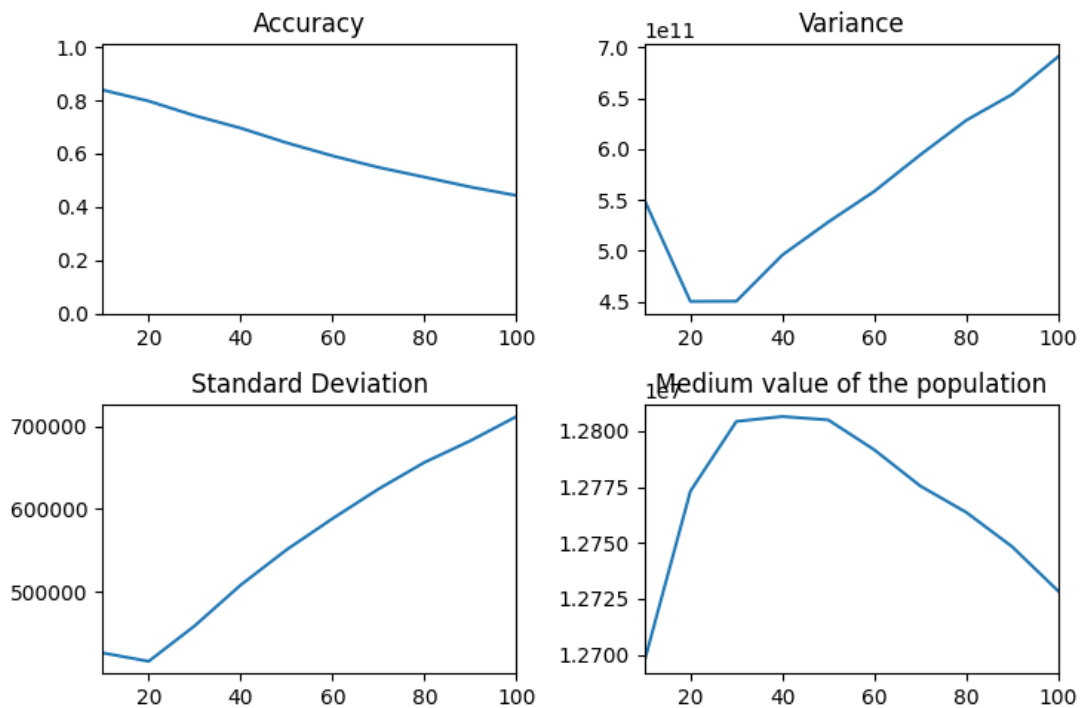


Figure 2: Statistics when we change the number of solutions

Pearson's correlation coefficient indicates that the correlation is not statistically significant, since the p-value result ($\simeq 1$) is well above the significance level.

From the graphs we can deduce that by increasing the number of solutions, the population, we introduce more variance to the population, which causes the accuracy to drop as well as the mean of the population value.

Modifying the maximum number of generations

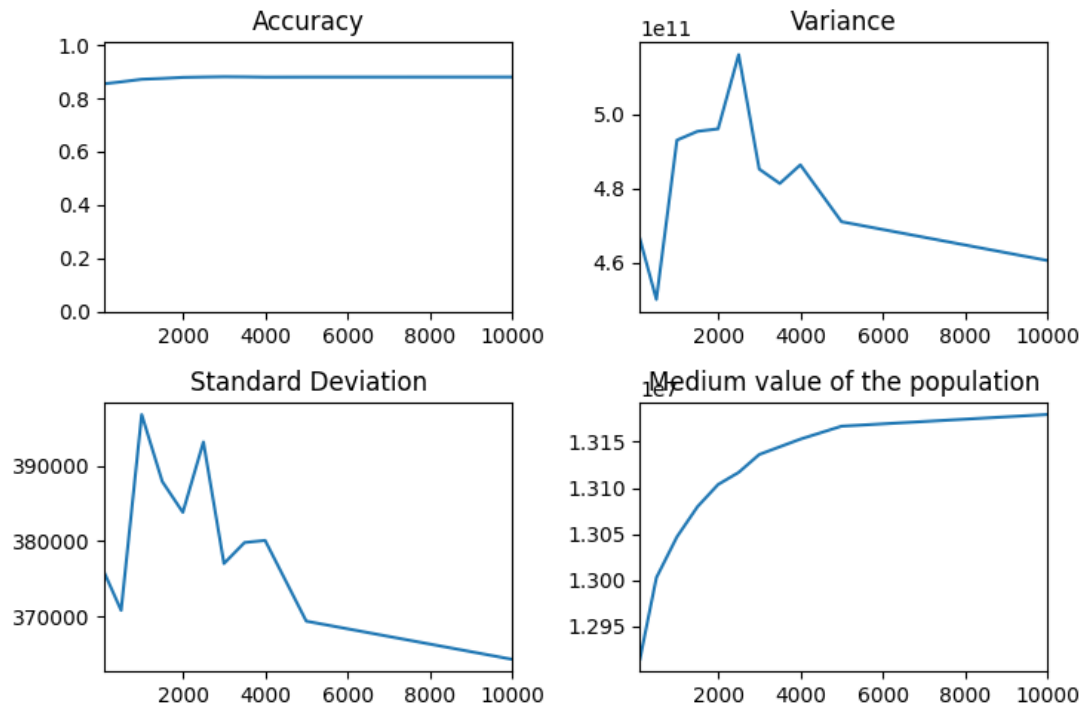


Figure 3: Statistics when we change the maximum number of generations

Pearson's correlation coefficient indicates that there is a moderate correlation (0.59) with a certainty of 94.8%, This indicates that both variables will have the same tendency, if the number of generations increases, the accuracy will increase. not hardly perceptible but it is undergoing slight increases in the graph.

The variance and standard deviation graphs show how they decrease as the maximum number of generations increases.

The population mean grows exponentially, this may be because the population is converging to the optimal solution.

Modifying the tournament size

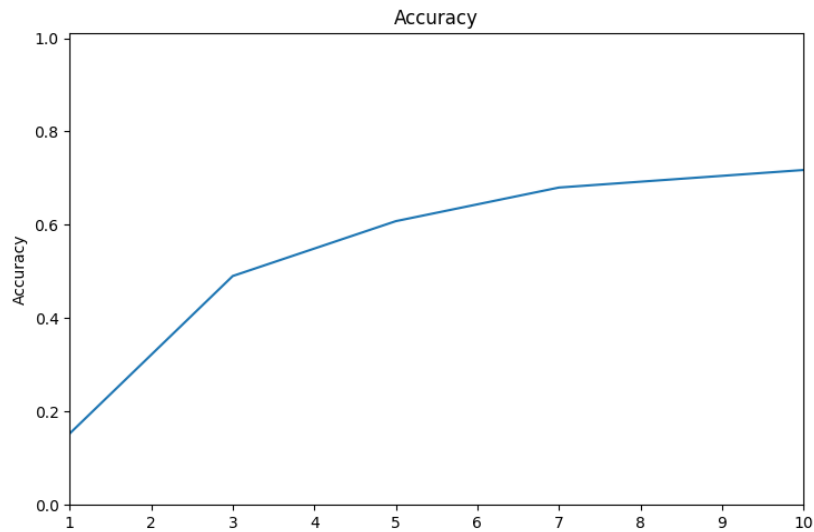


Figure 4: Statistics when we change the tournament size

For this section I have had some problems in representing the values of the mean, variance and standard deviation.

```
import numpy as np

....

mean = np.average(knapsack_values)
var = np.var(knapsack_values)
std = np.std(knapsack_values)
```

This code for this case provides a Runtime Warning and the results it gives are nan.

Pearson's correlation coefficient tells us that there is a strong correlation (0.88) with a certainty of 95.5%.

Modifying the crossover probability

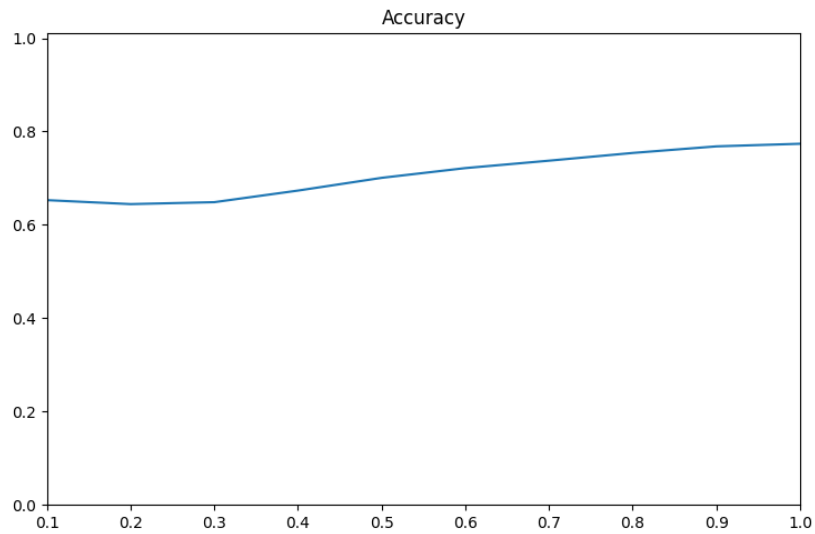


Figure 5: Statistics when we change the probability of crossover

In this section I have had the same problem as in the previous one when obtaining the graphs of the variance, standard deviation and mean of the population.

The Pearson correlation coefficient indicates that there is a strong correlation (0.97) with a certainty of 99.9%.

Modifying the mutation probability

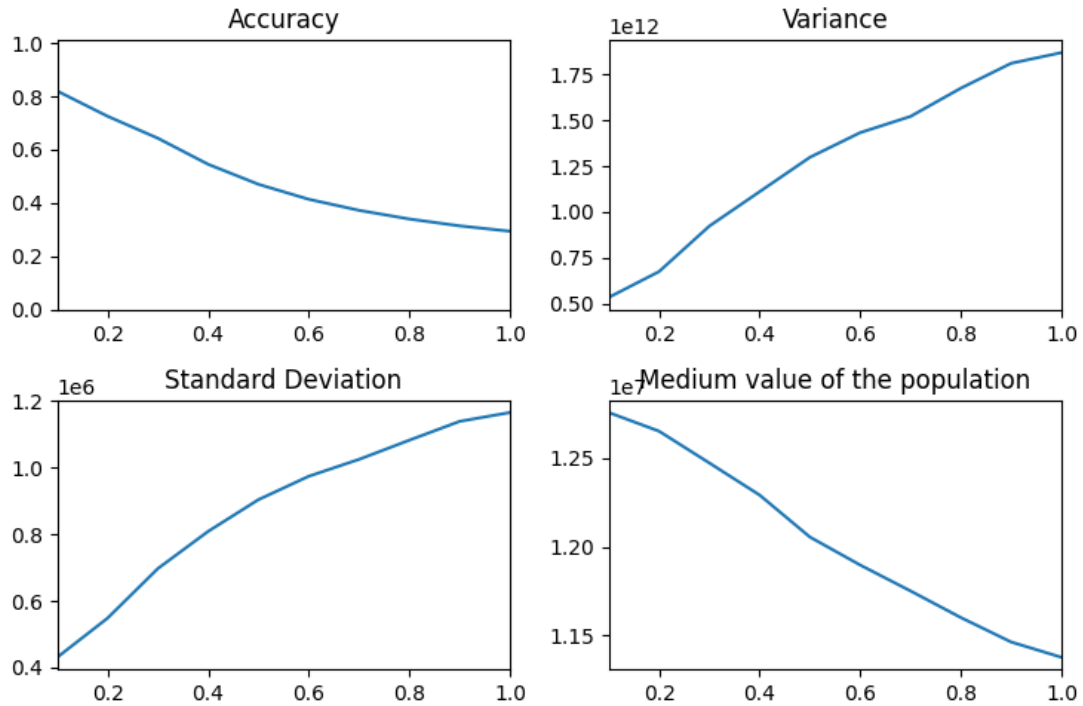


Figure 6: Statistics when we change the probability of mutation

As the probability of mutation increases, areas further away from the most optimal area will be explored with a higher probability. This causes the accuracy to decrease, as well as the mean value of the population. The opposite happens, the variance and standard deviation increase because we are including individuals that are very different from the population.

Pearson's correlation coefficient indicates that there is a strong correlation (-0.97) with a certainty of 99.9%. indicates that as one of the variables increases, the other decreases.

1.3 Conclusions of the behavior of the algorithm

Taking into account the analysis carried out we can conclude what would be the ideal combination of hyperparameters for our problem.

For the number of solutions we can say that the ideal value is 20, where the accuracy is around 80%, the variance and standard deviation reaches its minimum, and the population mean does not reach its maximum, but is very close. and the population mean does not reach its maximum, but it is very close.

For the maximum number of generations we can say that the higher the value the better, as this will minimise the variance and standard deviation functions and maximise the mean value of the population. Although the higher the value the better, we should be aware that these operations will require a lot of computational time, so we should establish a stopping criterion

that takes into account, for instance, how much the algorithm has improved over the last few iterations and if this improvement is insignificant, stop.

For the tournament size we would use a value of 10, as this gives us a higher accuracy.

For the crossover probability we will use 1 or a value close to one, as these have the highest accuracy.

For the mutation probability we would use a value the smaller the better, as this gives us a value of 80% accuracy, a high population mean value and a small value for variance and standard deviation.

1.4 Do you always get the best solution? Why? What does it depend on?

No, the algorithm may converge prematurely to an area with good results, but not the best, to avoid this we can either increase the mutation probability, as this ensures that we will explore other areas of the search space, or we can apply elitism.

It depends on the value assigned to the hyperparameters.

- 1.5 Modify the code to incorporate elitism. The best solution is kept in the elite and it is never lost until a new better one is obtained. This solution is the one that is returned at the end. Have you managed to improve? Why?

Modifying the number of solutions

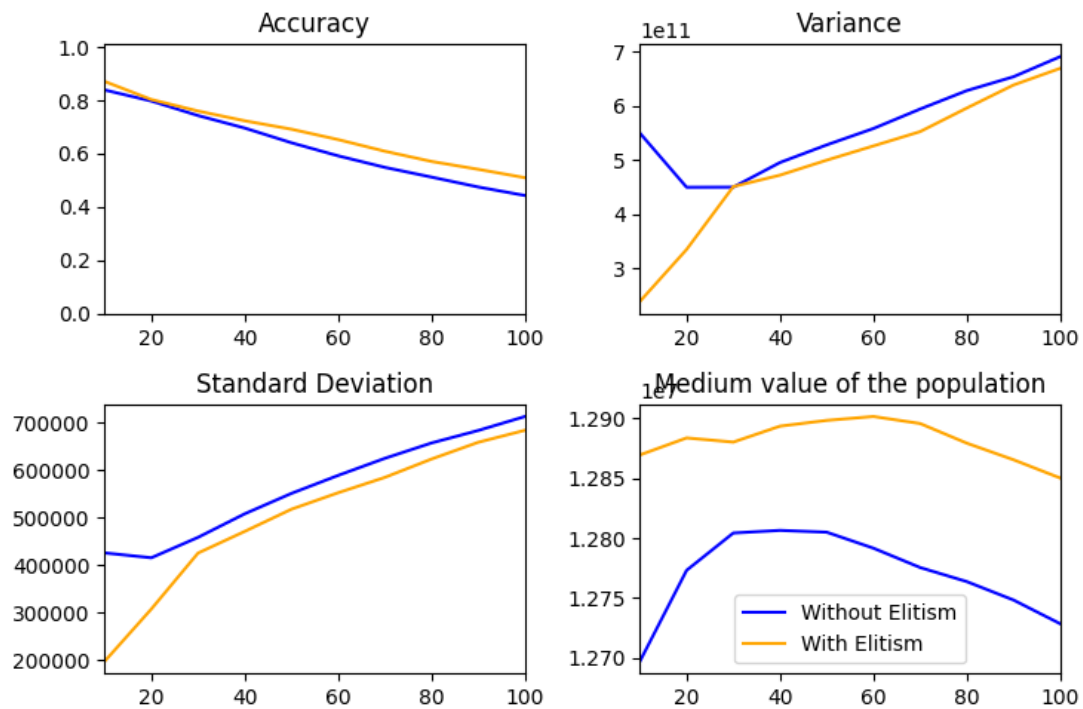


Figure 7: Statistics when we change the number of solutions and use elitism

Modifying the maximum number of generations

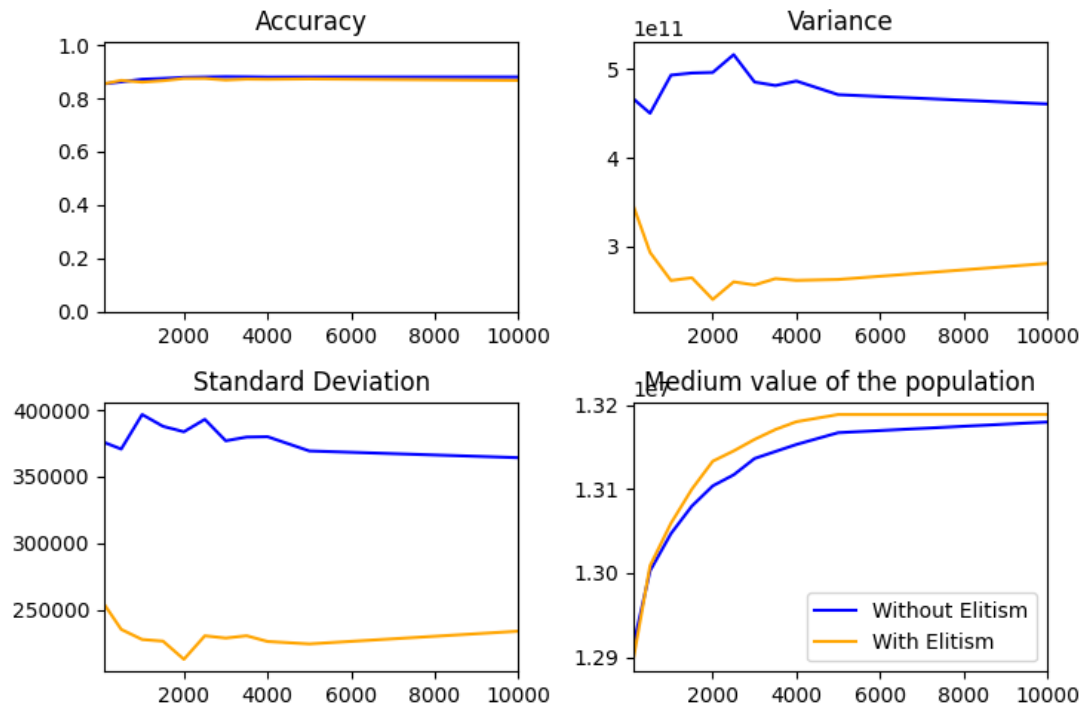


Figure 8: Statistics when we change the maximum number of generations and use elitism

Modifying the tournament size

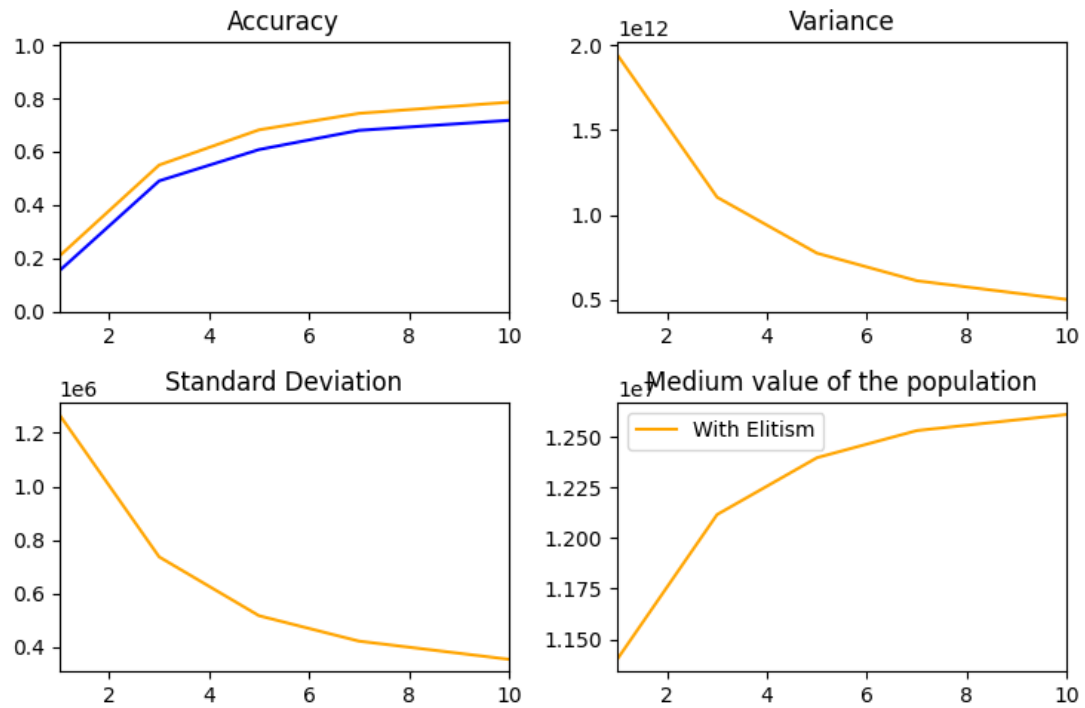


Figure 9: Statistics when we change the tournament size and use elitism

Modifying the crossover probability

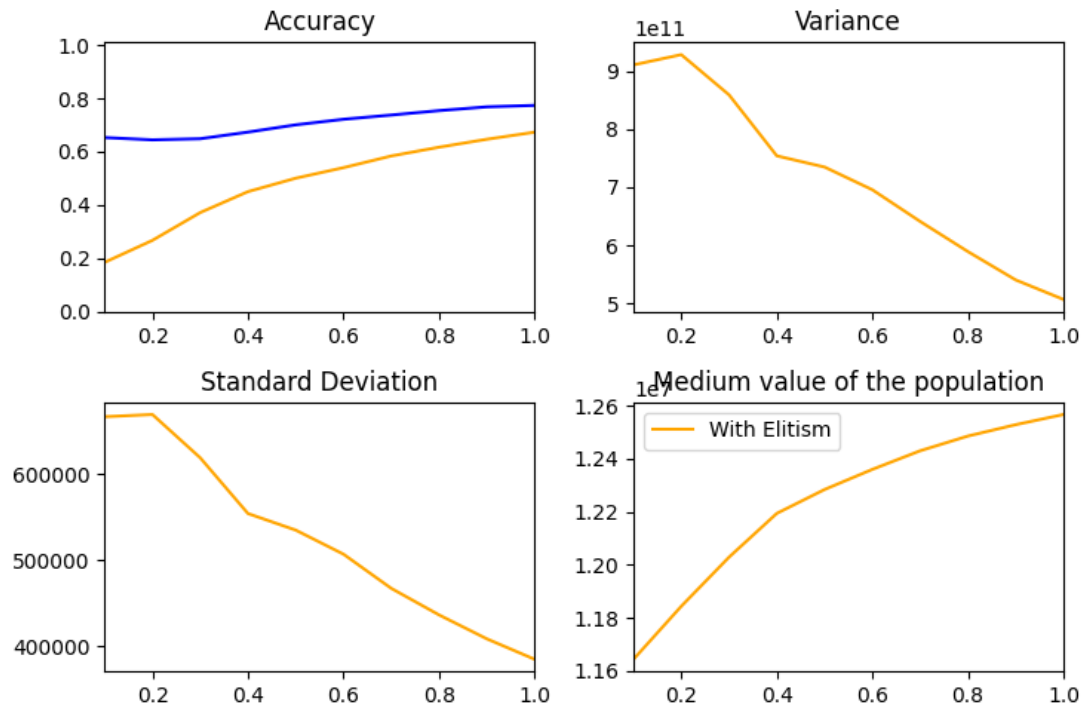


Figure 10: Statistics when we change the probability of crossover and use elitism

Modifying the mutation probability

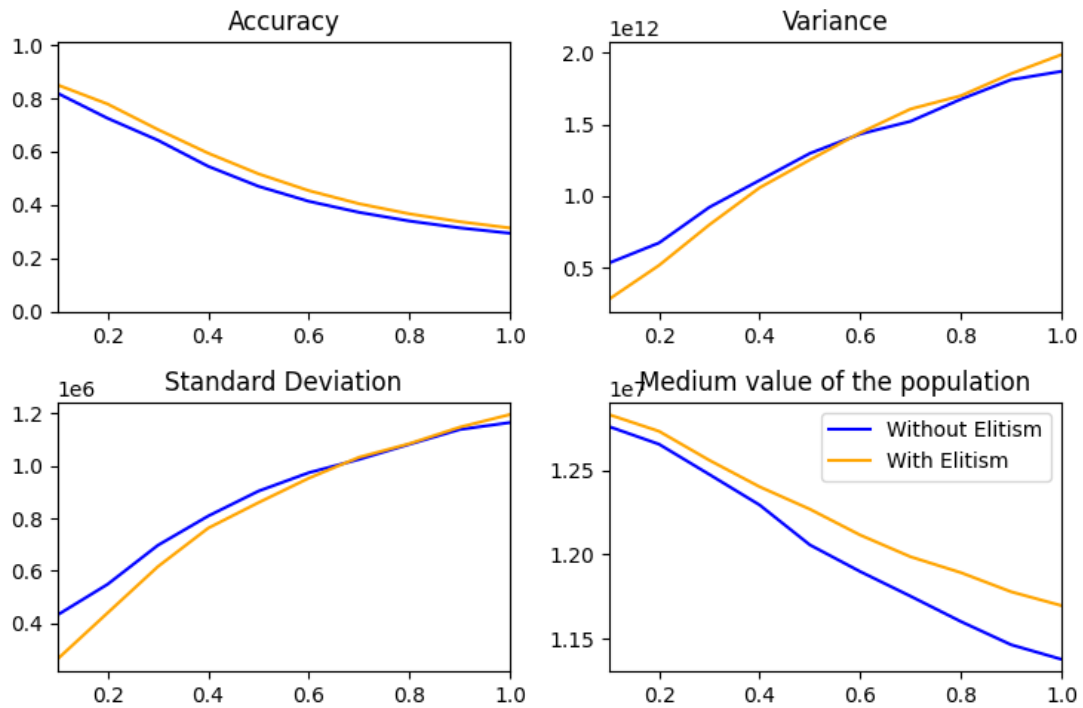


Figure 11: Statistics when we change the probability of mutation and use elitism

1.6 Conclusions of the behavior of the algorithm with elitism

In general, the accuracy value increases slightly, as does the mean value of the knapsack, and the variance and standard deviation curves smooth out. and the variance and standard deviation curves are smoothed.

So, yes, the algorithm has been improved by applying elitism.

1.7 So far, we have started with valid solutions. Change it so that any solution can be generated. Does it affect the final performance? Analyze this issue.

Modifying the number of solutions

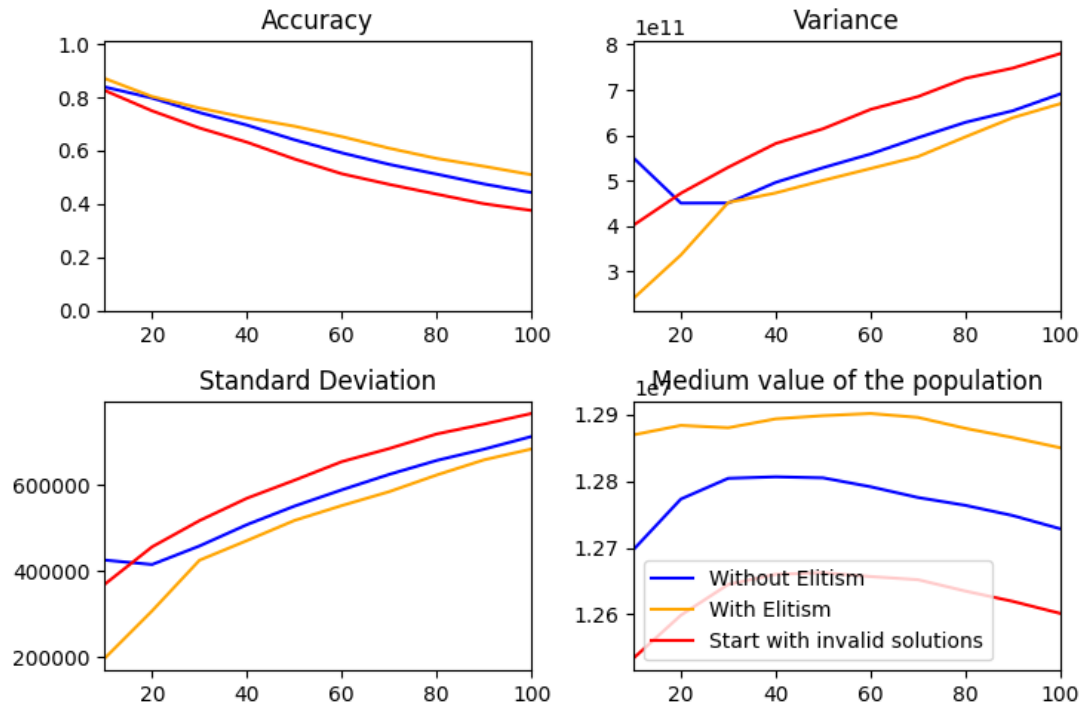


Figure 12: Statistics when we change the number of solutions and start with invalid solutions

Modifying the maximum number of generations

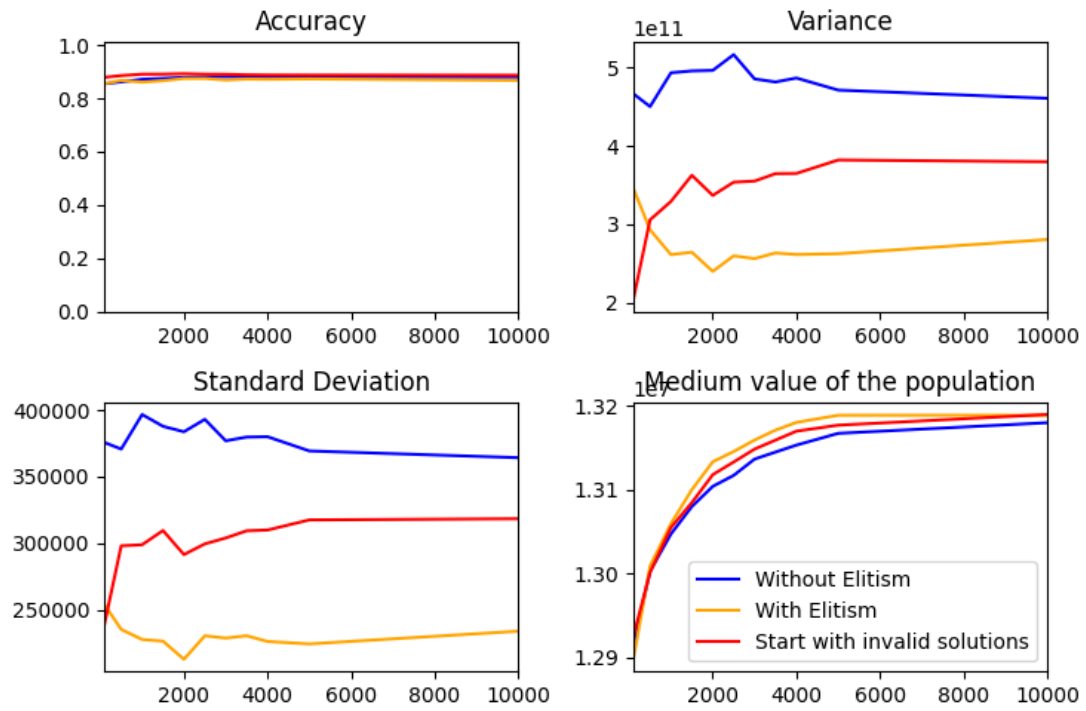


Figure 13: Statistics when we change the maximum number of generations and start with invalid solutions

Modifying the tournament size

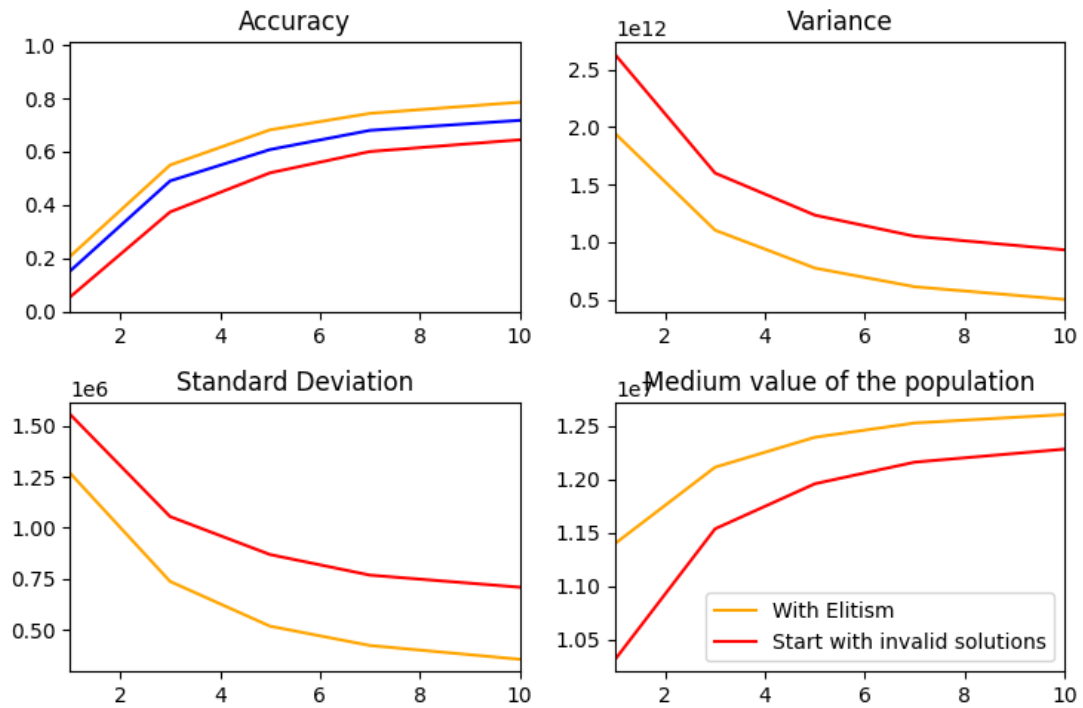


Figure 14: Statistics when we change the tournament size and start with invalid solutions

Modifying the crossover probability

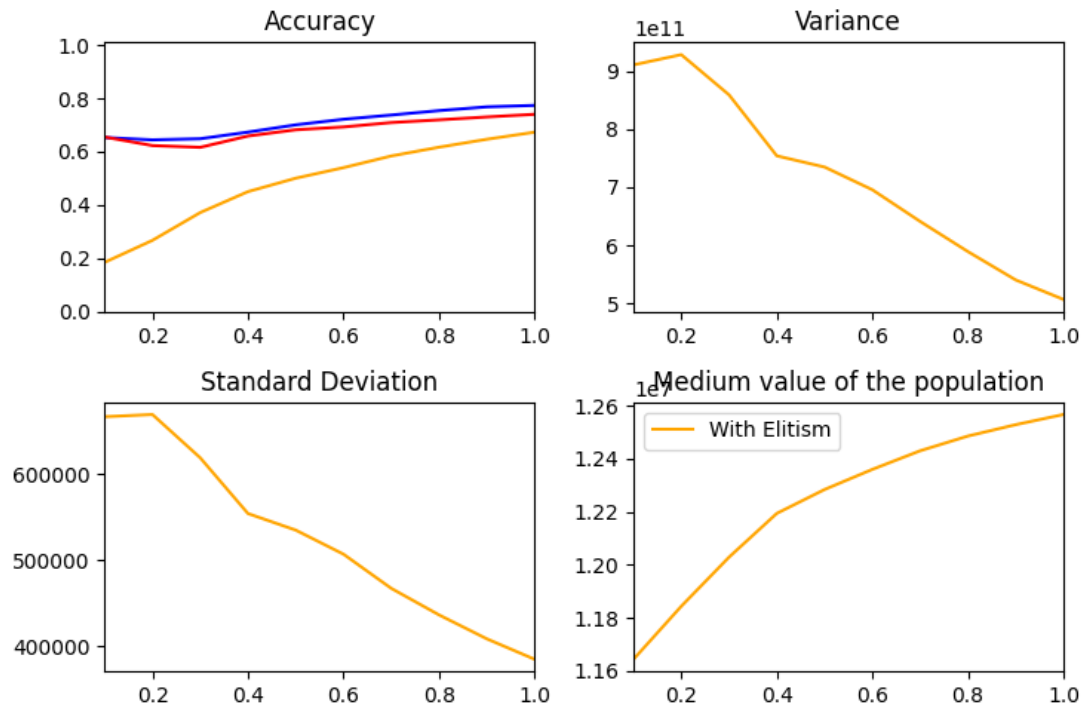


Figure 15: Statistics when we change the probability and start with invalid solutions

Modifying the mutation probability

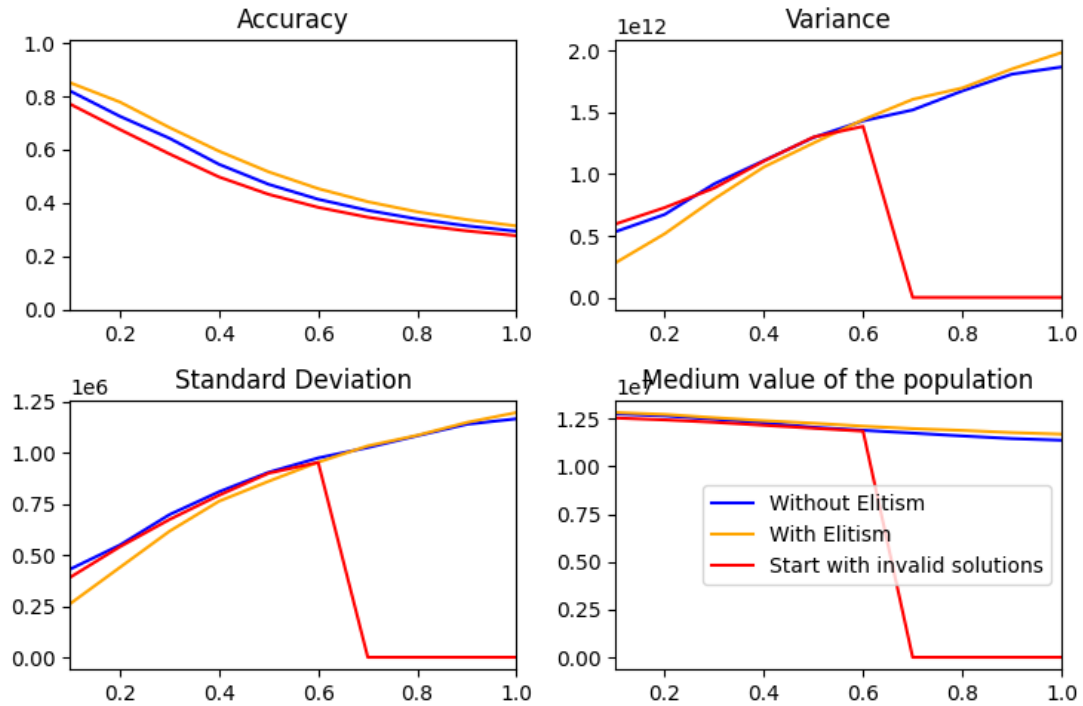


Figure 16: Statistics when we change the probability of mutation and start with invalid solutions

The graph falls down to 0 because the results for that probabilities are NaN.

1.8 Conclusions of the behavior of the algorithm with elitism

In general, the accuracy is smaller than the accuracy starting with valid solutions or using elitism, as same as the average value of the knapsack. On the other hand using invalid solutions increase the variance and the standard deviation.