# MDL Genetic Algorithms Report

Team number : 46
Team V
Team members :
1.varun Chowdary(2019101114).
2.Abhijith chelpuri(2019111036).

The project is applying genetic algorithm on the overfit vector given to obtain a better vector such that it's MSE is lower on train, validation and test data.

## Getting started:

The genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

The main code for the project is in main.py file.

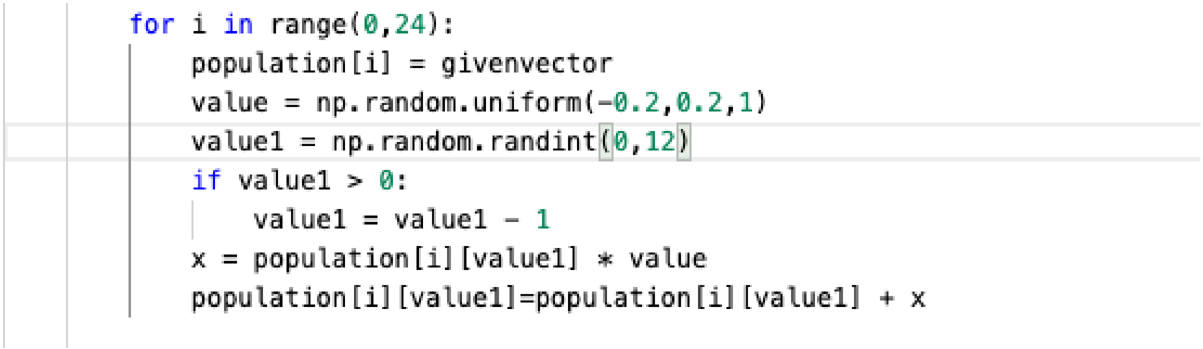
This is the final code after so many changes in the code some are minor and some are major.

# BASIC SUMMARY OF THE ALGORITHM:

## INTIALIZING POPULATION:

In the start we have intialized the population from the given overfit randomly around the given vector as shown:

We have initialized the population size to 24.Which remains constant throughout the generations.(Basic rule of genetic algorithm).



```
for i in range(0,24):
    population[i] = givenvector
    value = np.random.uniform(-0.2,0.2,1)
    value1 = np.random.randint(0,12)
    if value1 > 0:
        value1 = value1 - 1
    x = population[i][value1] * value
    population[i][value1]=population[i][value1] + x
```

I have done Value1-1 because when I assigned the value between 0 and 11 from my observations most of the time the first index in the vector is not getting changed so I did this.

The genetic Algorithm selects 6 Fittest parents from the population.

Number of Parents selected From population: 6.

Next generation population and how are they obtained:

1.From the Crossover of Previous generation parents: 9.

2.From the crossover of the vectors obtained from crossover : 9.

3.From the mutation of Parents in previous generations : 3.

4.From the crossover of mutations obtained in previous generation: 3.

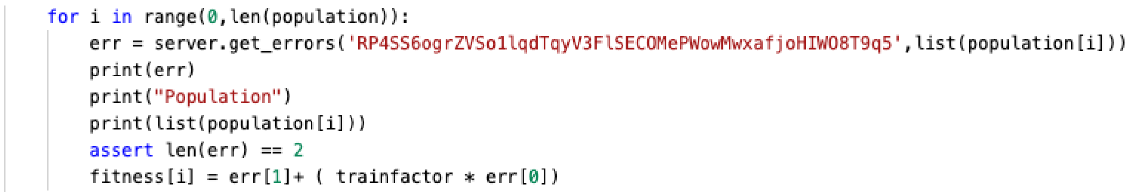
The Above parameters changes throughout the Process of Getting the best vector.

Then we calculated Fitness of the each Vector in the population Then Choose as above told 6 parents.

Then we implemented The crossover and mutation of the parent and the vectors obtained in the before mutation and crossover.

In algorithm we implemented something different from normal genetic algorithm.

## CALCULATING ERRORS FOR THE POPULATION:

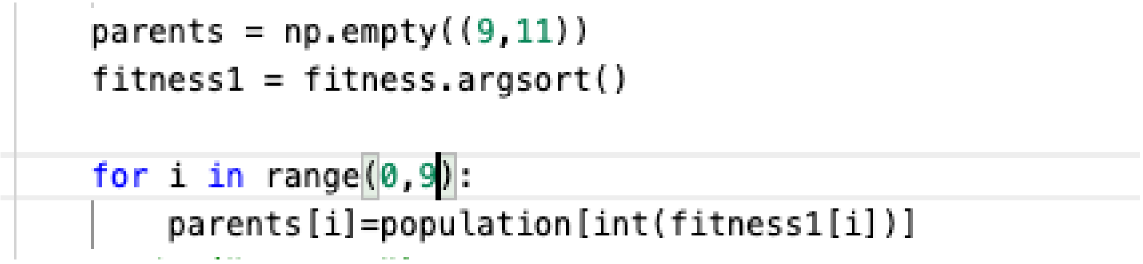


```
for i in range(0,len(population)):
    err = server.get_errors('RP4SS6ogrZVSo1lqdTqyV3FlSECOMePWowMwxafjoHIWO8T9q5',list(population[i]))
    print(err)
    print("Population")
    print(list(population[i]))
    assert len(err) == 2
    fitness[i] = err[1]+ ( trainfactor * err[0])
```

We put all the errors in the fitness 2-D array which has both validation error and the train error in it in the index (i,o) and (i,1)

## CREATING PARENT POPULATION:

What we have done here is simply is taking the population with least errors the argsort() function does this in the ascending order and then we placed the arrays in the parents array.



```
parents = np.empty((9,11))
fitness1 = fitness.argsort()

for i in range(0,9):
    parents[i]=population[int(fitness1[i])]
```

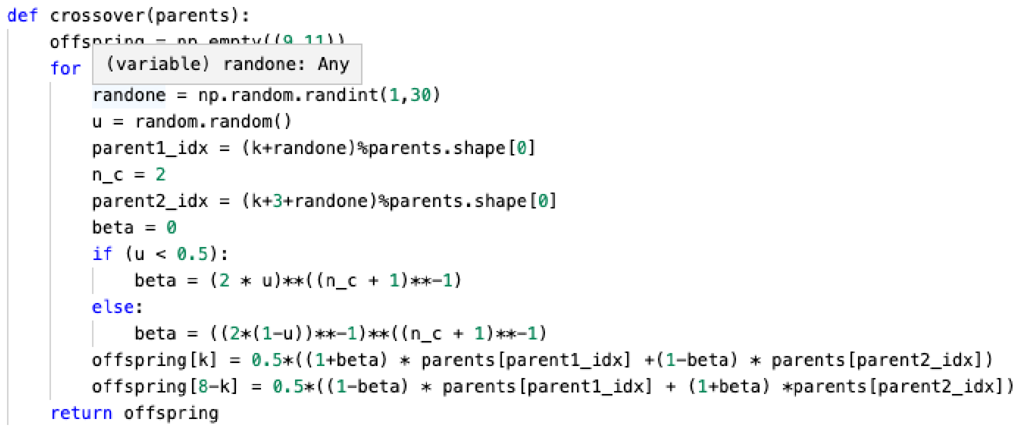
## CREATING CROSSOVER FOR POPULATION OF NEXT GENERATION:

This is what we have done for crossover for the parents i.e population which have the least error.

offspring_crossover =  crossover(parents)

population[0:9,:]=offspring_crossover[0:9,:]

There is a crossover which functions as below:



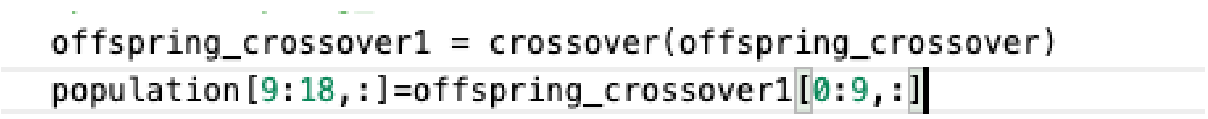
```
def crossover(parents):
    offspring = np.empty((9,11))
    for (variable) randone: Any
        randone = np.random.randint(1,30)
        u = random.random()
        parent1_idx = (k+randone)%parents.shape[0]
        n_c = 2
        parent2_idx = (k+3+randone)%parents.shape[0]
        beta = 0
        if (u < 0.5):
            beta = (2 * u)**((n_c + 1)**-1)
        else:
            beta = ((2*(1-u))**-1)**((n_c + 1)**-1)
        offspring[k] = 0.5*((1+beta) * parents[parent1_idx] +(1-beta) * parents[parent2_idx])
        offspring[8-k] = 0.5*((1-beta) * parents[parent1_idx] + (1+beta) *parents[parent2_idx])
    return offspring
```

The crossover is done as above we have selected parents randomly from parents and then performed the **simulated binary crossover**.

## CROSSOVER OF THE CHILDREN OBTAINED ABOVE:

We have performed the crossover of the children obtained in the above generation to make the population more better and to reduce the requests in a way.

It is done as below



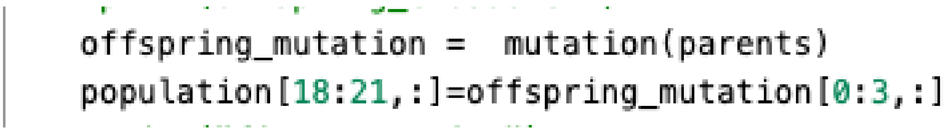
```
offspring_crossover1 = crossover(offspring_crossover)
population[9:18,:]=offspring_crossover1[0:9,:]
```

And the crossover function is same as above.

## MUTATION OF THE PARENTS OBTAINED FROM POPULATION:

We have performed mutation on the parents we got from the population and then performed mutation on those we changes the population of mutation and how far we go from the parents in the code several times.

AS shown below:

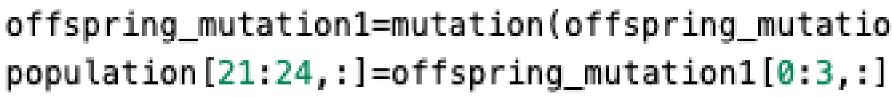


```
offspring_mutation =  mutation(parents)
population[18:21,:]=offspring_mutation[0:3,:]
```

The size of the mutation varied from time to time in our process to get the best vectors.

## MUTATION OF THE MUTATION VECTORS OBTAINED ABOVE:

We have performed double mutation on the already mutated vectors this is an improvement of the first code we have written at some stage this is done so that we have some vectors different from the parent vectors so that we have a wide view of the vectors in a single generation. It is done as Shown below.



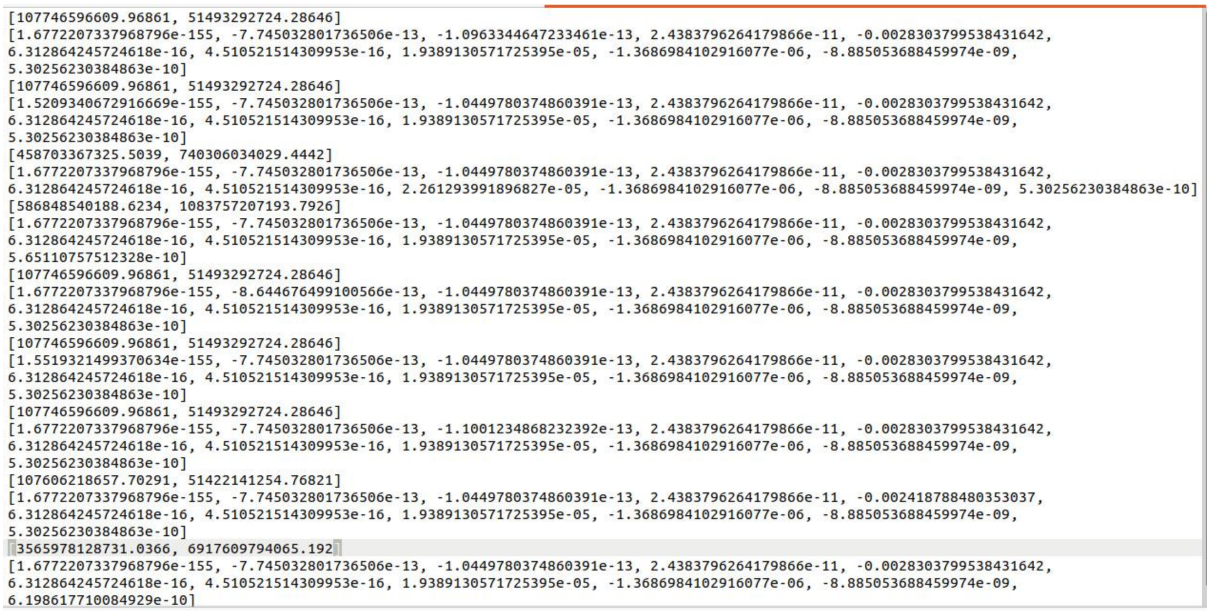
```
offspring_mutation1=mutation(offspring_mutation)
population[21:24,:]=offspring_mutation1[0:3,:]
```

## THE ITERATION DIAGRAMS ARE AS FOLLOWS:

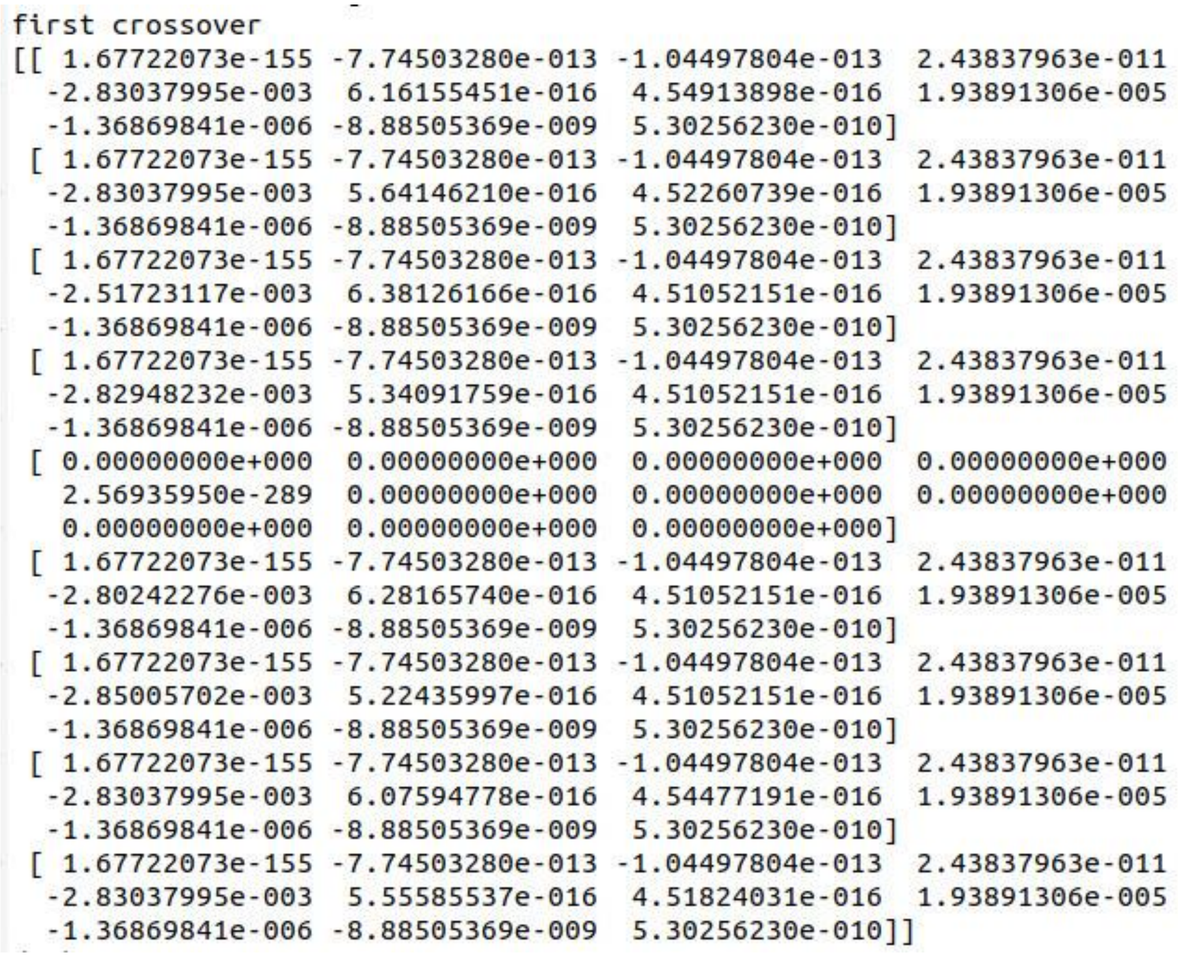
Details of the pics shown below and their explanation

- The first pic in every iteration shows the parents we selected for the crossover from the population
- The second pic shows the first crossover of the parents in above pic
- The third pic shows the crossover of the crossovervectors obtained in second pic.
- The fourth pic shows the mutation of parents obtained from parents in first pic.
- The last pic shows the mutation of the mutated vectors in the fourth pic.
-

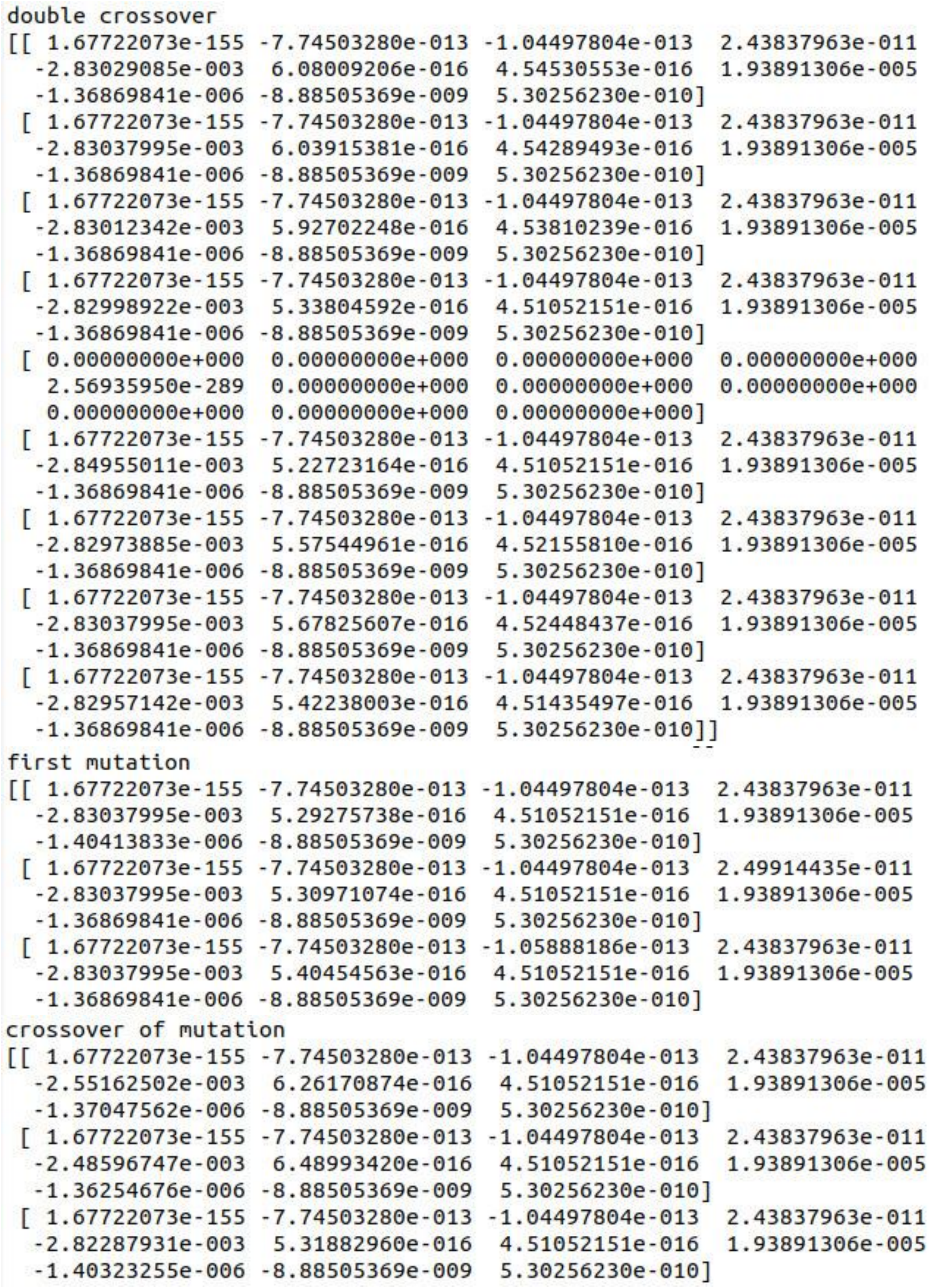
## FIRST ITERATION:



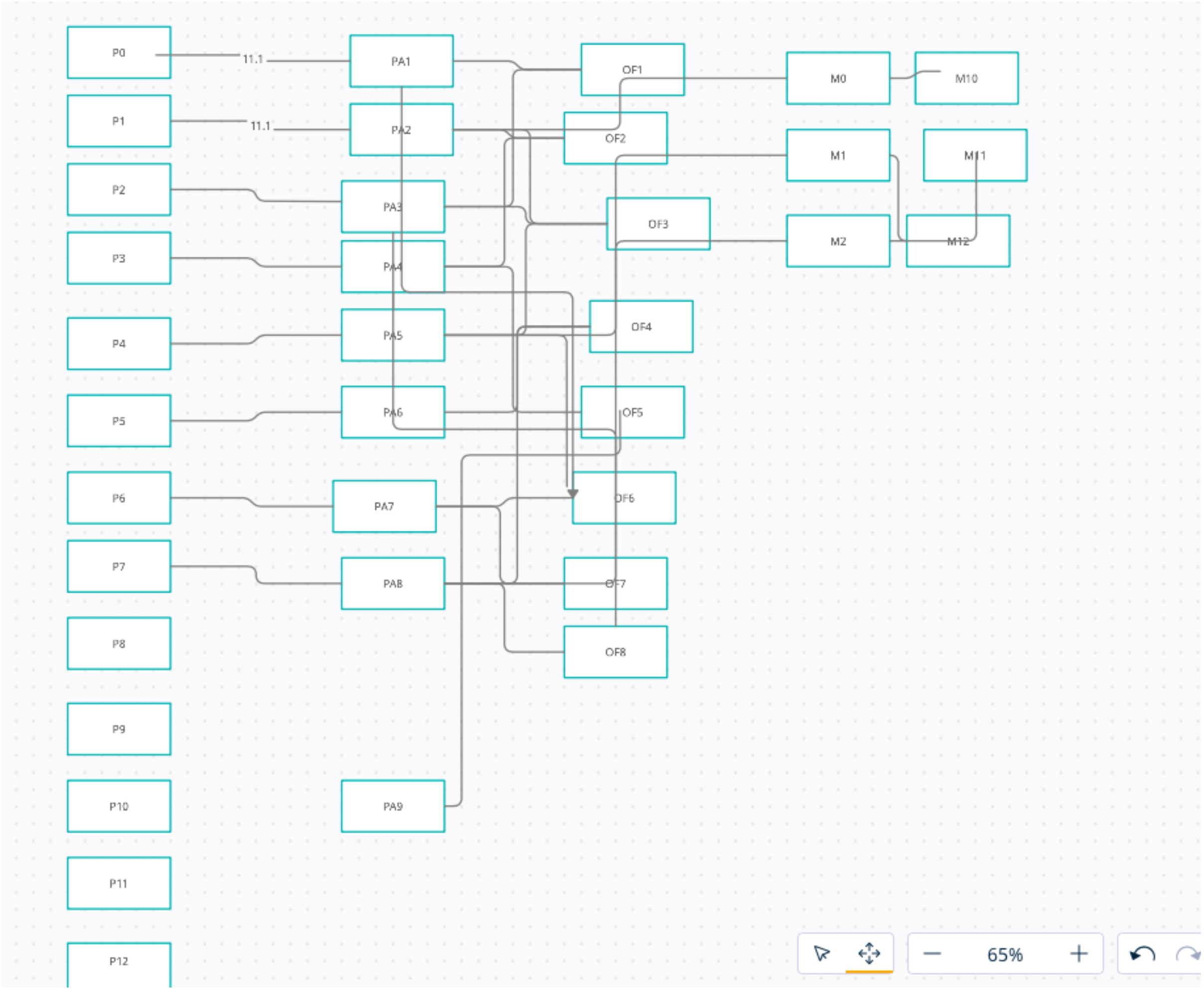
```
[107746596609.96861, 51493292724.28646]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0963344647233461e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[107746596609.96861, 51493292724.28646]
[1.5209340672916669e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[458703367325.5039, 740306034029.4442]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 2.261293991896827e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[586848540188.6234, 1083757207193.7926]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.65110757512328e-10]
[107746596609.96861, 51493292724.28646]
[1.6772207337968796e-155, -8.644676499100566e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[107746596609.96861, 51493292724.28646]
[1.5519321499370634e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[107746596609.96861, 51493292724.28646]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.1001234868232392e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[107606218657.70291, 51422141254.76821]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.002418788480353037,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
5.30256230384863e-10]
[3565978128731.0366, 6917609794065.192]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
6.312864245724618e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09,
6.198617710084929e-10]
```



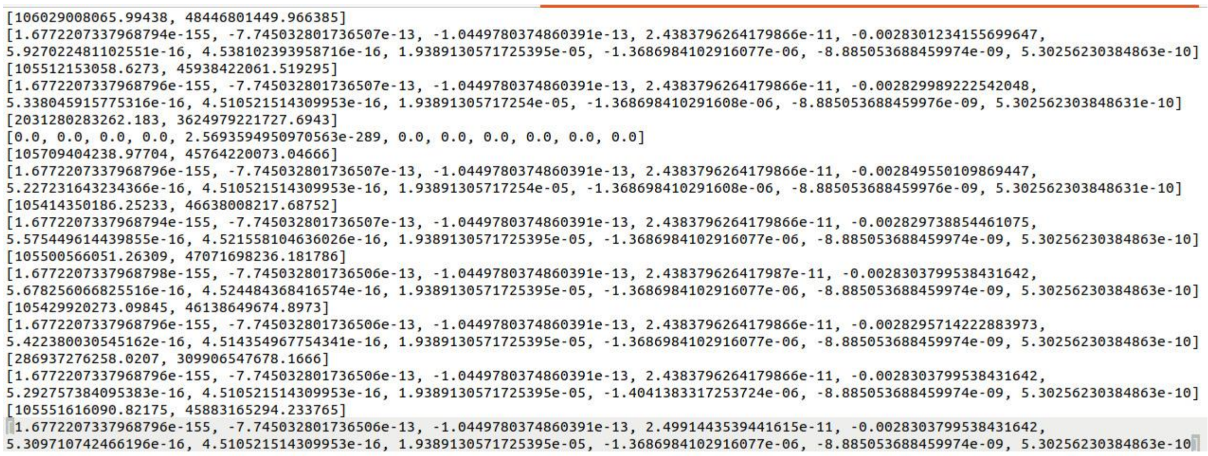
```
first crossover
[[ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  6.16155451e-016  4.54913898e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  5.64146210e-016  4.52260739e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.51723117e-003  6.38126166e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82948232e-003  5.34091759e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.80242276e-003  6.28165740e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.85005702e-003  5.22435997e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  6.07594778e-016  4.54477191e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  5.55585537e-016  4.51824031e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]]
```



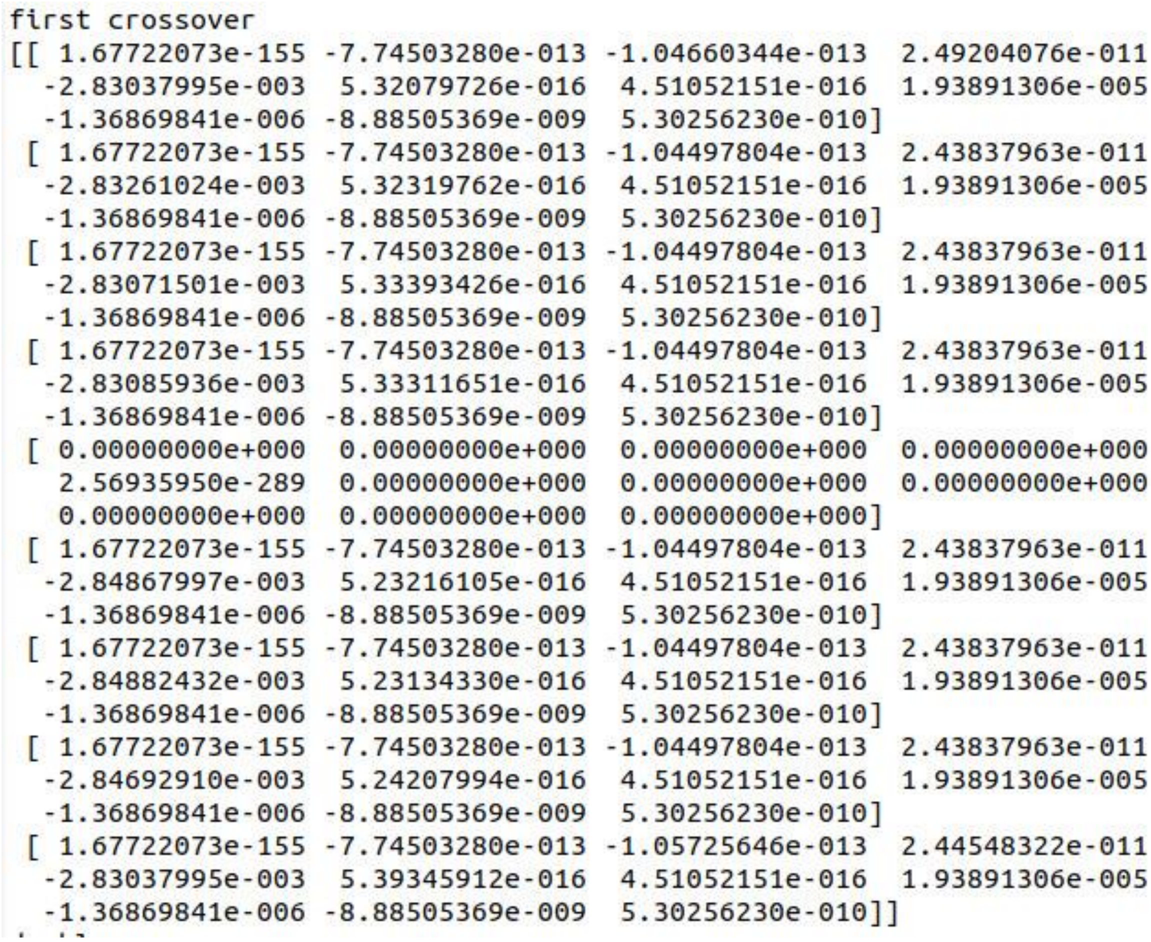
```
double crossover
[[ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83029085e-003  6.08009206e-016  4.54530553e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  6.03915381e-016  4.54289493e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83012342e-003  5.92702248e-016  4.53810239e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82998922e-003  5.33804592e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84955011e-003  5.22723164e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82973885e-003  5.57544961e-016  4.52155810e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  5.67825607e-016  4.52448437e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82957142e-003  5.42238003e-016  4.51435497e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]]
first mutation
[[ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83037995e-003  5.29275738e-016  4.51052151e-016  1.93891306e-005
  -1.40413833e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.49914435e-011
  -2.83037995e-003  5.30971074e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.05888186e-013  2.43837963e-011
  -2.83037995e-003  5.40454563e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
crossover of mutation
[[ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.55162502e-003  6.26170874e-016  4.51052151e-016  1.93891306e-005
  -1.37047562e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.48596747e-003  6.48993420e-016  4.51052151e-016  1.93891306e-005
  -1.36254676e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82287931e-003  5.31882960e-016  4.51052151e-016  1.93891306e-005
  -1.40323255e-006 -8.88505369e-009  5.30256230e-010]
```



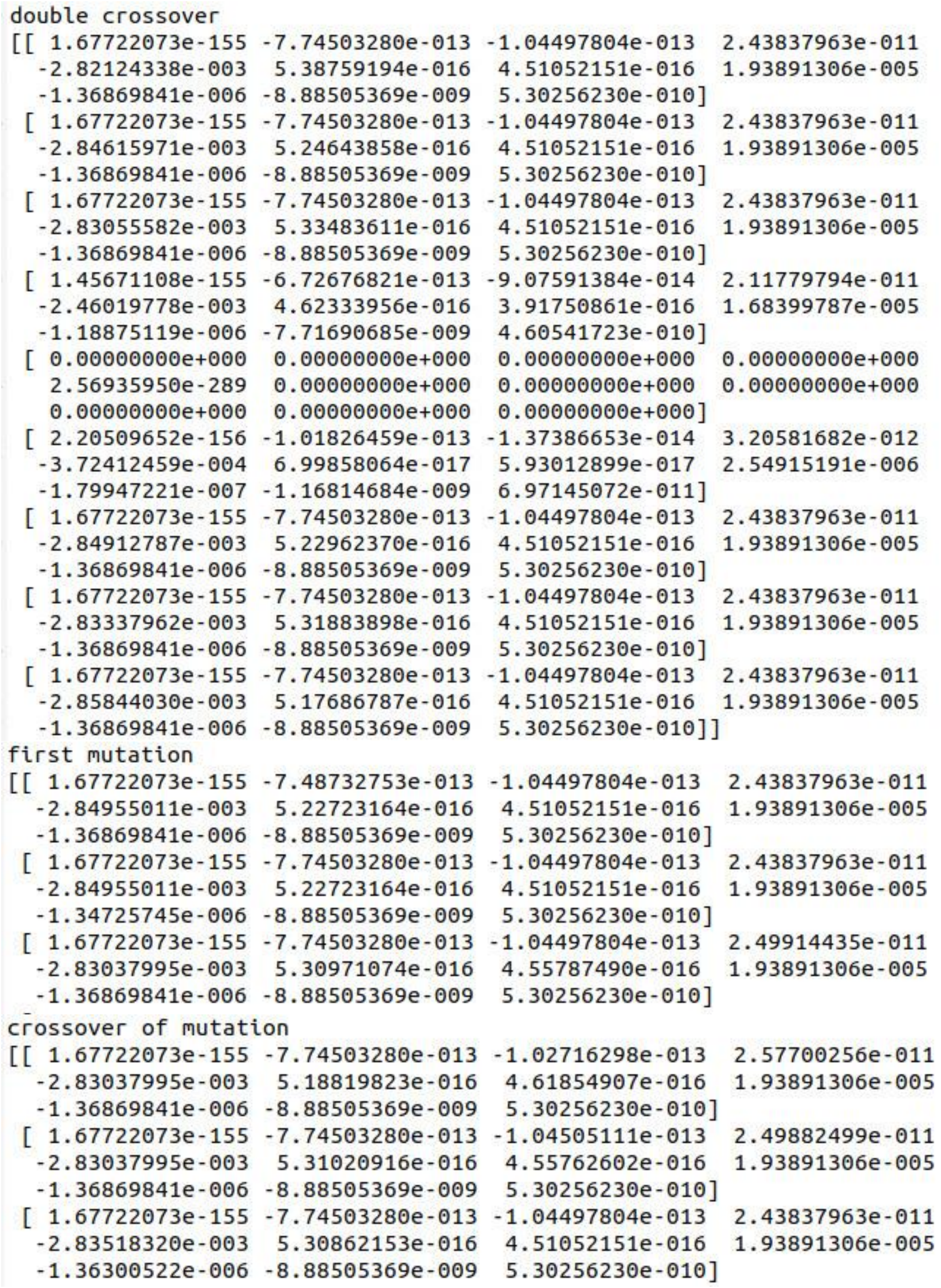
# SECOND ITERATION:



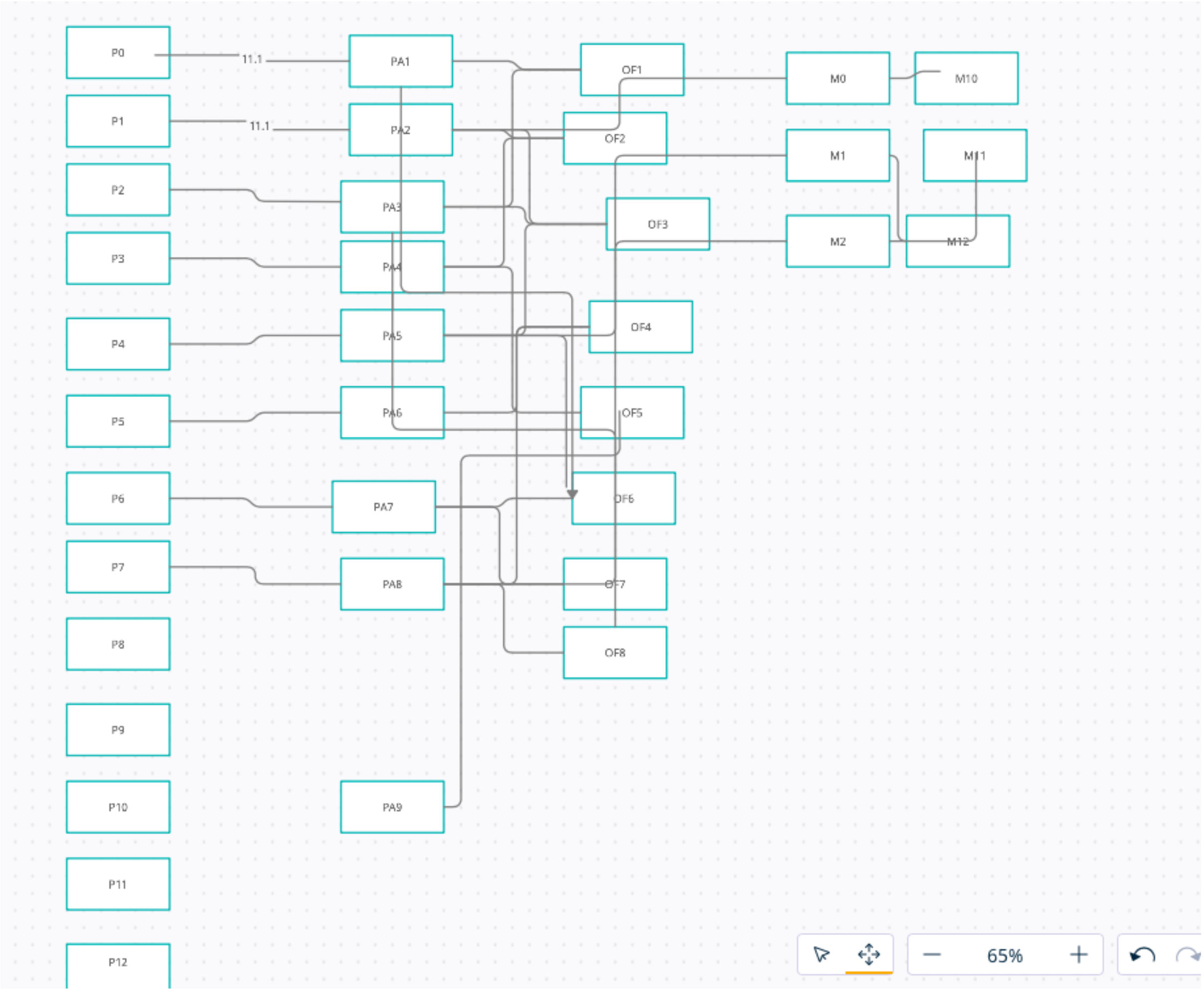
```
[106029008065.99438, 48446801449.966385]
[1.6772207337968794e-155, -7.745032801736507e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028301234155699647,
5.927022481102551e-16, 4.538102393958716e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[105512153058.6273, 45938422061.519295]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.00282989222542048,
5.338045915775316e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.368698410291608e-06, -8.885053688459976e-09, 5.302562303848631e-10]
[2031280283262.183, 3624979221727.6943]
[0.0, 0.0, 0.0, 0.0, 2.5693594950970563e-289, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[105709404238.97704, 45764220073.04666]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.002849550109869447,
5.227231643234366e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.368698410291608e-06, -8.885053688459976e-09, 5.302562303848631e-10]
[105414350186.25233, 46638008217.68752]
[1.6772207337968794e-155, -7.745032801736507e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.002829738854461075,
5.575449614439855e-16, 4.521558104636026e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[105500566051.26309, 47071698236.181786]
[1.6772207337968798e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.438379626417987e-11, -0.0028303799538431642,
5.678256066825516e-16, 4.524484368416574e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[105429920273.09845, 46138649674.8973]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028295714222883973,
5.422380030545162e-16, 4.514354967754341e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[286937276258.0207, 309906547678.1666]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.0028303799538431642,
5.292757384095383e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.4041383317253724e-06, -8.885053688459974e-09, 5.30256230384863e-10]
[105551616090.82175, 45883165294.233765]
[1.6772207337968796e-155, -7.745032801736506e-13, -1.0449780374860391e-13, 2.4991443539441615e-11, -0.0028303799538431642,
5.309710742466196e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.30256230384863e-10]
```



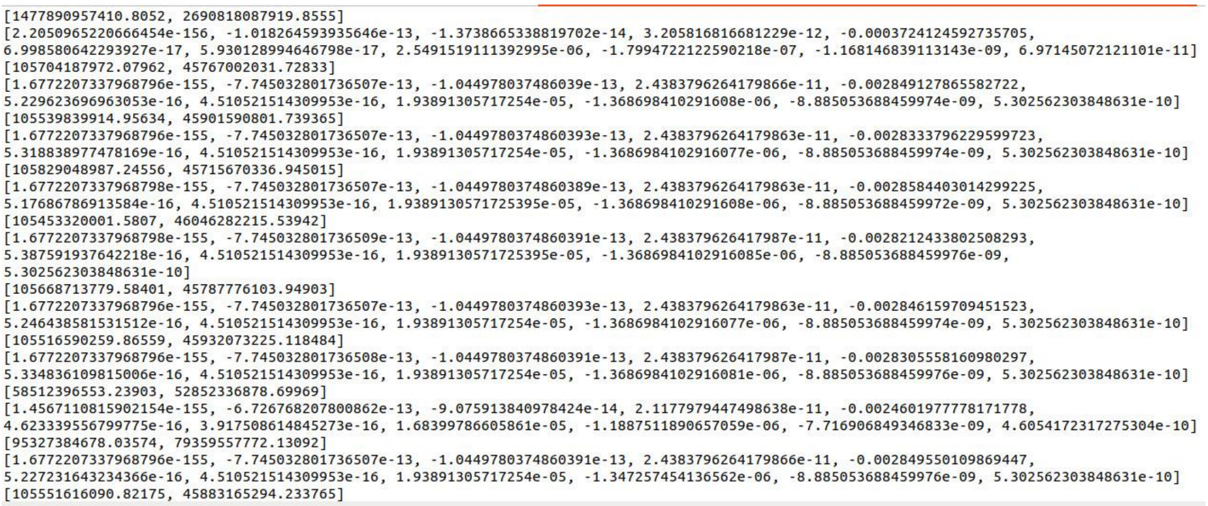
```
first crossover
[[ 1.67722073e-155 -7.74503280e-013 -1.04660344e-013  2.49204076e-011
  -2.83037995e-003  5.32079726e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83261024e-003  5.32319762e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83071501e-003  5.33393426e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83085936e-003  5.33311651e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84867997e-003  5.23216105e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84882432e-003  5.23134330e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84692910e-003  5.24207994e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.05725646e-013  2.44548322e-011
  -2.83037995e-003  5.39345912e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]]
```



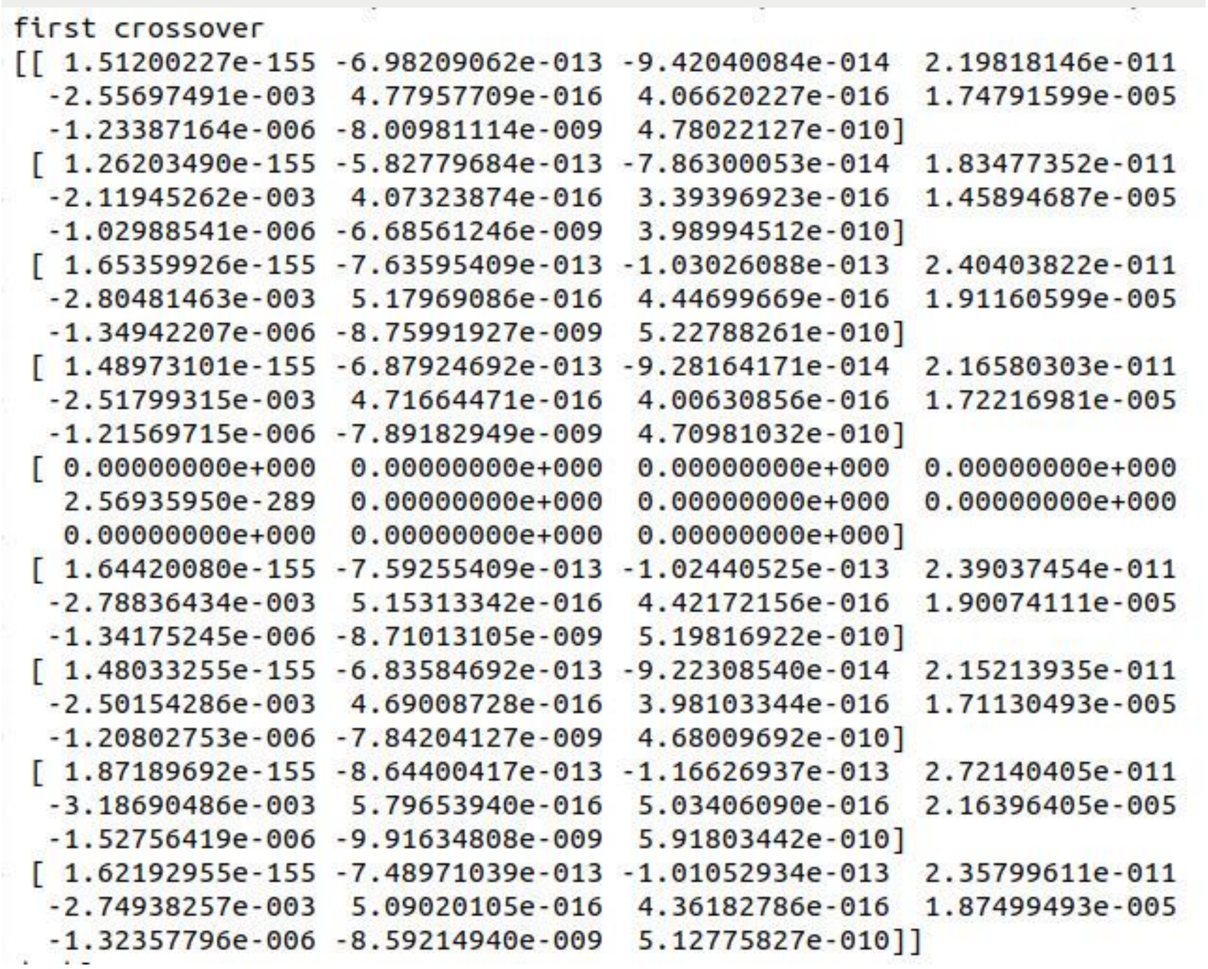
```
double crossover
[[ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.82124338e-003  5.38759194e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84615971e-003  5.24643858e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83055582e-003  5.33483611e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.45671108e-155 -6.72676821e-013 -9.07591384e-014  2.11779794e-011
  -2.46019778e-003  4.62333956e-016  3.91750861e-016  1.68399787e-005
  -1.18875119e-006 -7.71690685e-009  4.60541723e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 2.20509652e-156 -1.01826459e-013 -1.37386653e-014  3.20581682e-012
  -3.72412459e-004  6.99858064e-017  5.93012899e-017  2.54915191e-006
  -1.79947221e-007 -1.16814684e-009  6.97145072e-011]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84912787e-003  5.22962370e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83337962e-003  5.31883898e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.85844030e-003  5.17686787e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]]
first mutation
[[ 1.67722073e-155 -7.48732753e-013 -1.04497804e-013  2.43837963e-011
  -2.84955011e-003  5.22723164e-016  4.51052151e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.84955011e-003  5.22723164e-016  4.51052151e-016  1.93891306e-005
  -1.34725745e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.49914435e-011
  -2.83037995e-003  5.30971074e-016  4.55787490e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
crossover of mutation
[[ 1.67722073e-155 -7.74503280e-013 -1.02716298e-013  2.57700256e-011
  -2.83037995e-003  5.18819823e-016  4.61854907e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04505111e-013  2.49882499e-011
  -2.83037995e-003  5.31020916e-016  4.55762602e-016  1.93891306e-005
  -1.36869841e-006 -8.88505369e-009  5.30256230e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83518320e-003  5.30862153e-016  4.51052151e-016  1.93891306e-005
  -1.36300522e-006 -8.88505369e-009  5.30256230e-010]
```



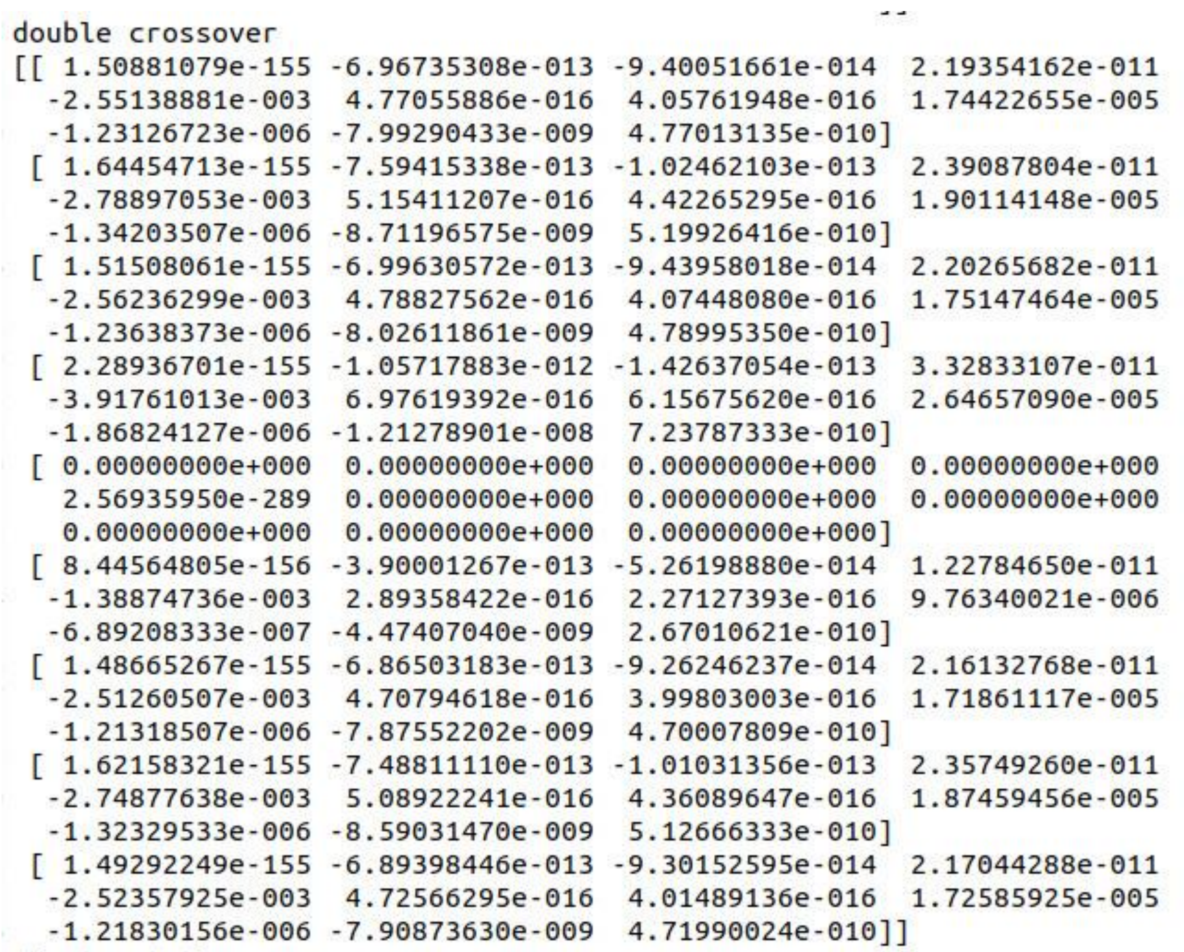
# THIRD ITERATION:



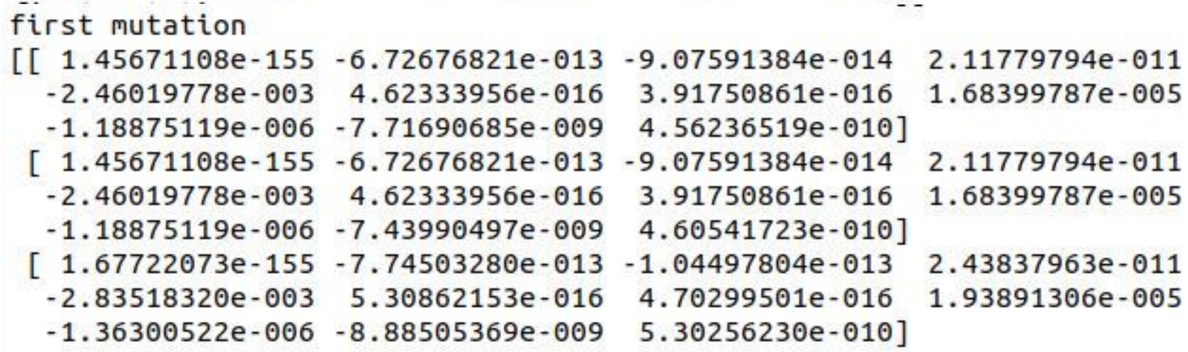
```
[1477890957410.8052, 2690818087919.8555]
[2.2050965220666454e-156, -1.018264593935646e-13, -1.3738665338819702e-14, 3.205816816681229e-12, -0.0003724124592735705,
6.998580642293927e-17, 5.930128994646798e-17, 2.5491519111392995e-06, -1.7994722122590218e-07, -1.168146839113143e-09, 6.97145072121101e-11]
[105704187972.07962, 45767002031.72833]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.044978037486039e-13, 2.4383796264179866e-11, -0.002849127865582722,
5.229623696963053e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.368698410291608e-06, -8.885053688459974e-09, 5.302562303848631e-10]
[105539839914.95634, 45901590801.739365]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.0449780374860393e-13, 2.4383796264179863e-11, -0.0028333796229599723,
5.318838977478169e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.302562303848631e-10]
[105829048987.24556, 45715670336.945015]
[1.6772207337968798e-155, -7.745032801736507e-13, -1.0449780374860389e-13, 2.4383796264179863e-11, -0.002858440301429925,
5.17686786913584e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.368698410291608e-06, -8.885053688459972e-09, 5.302562303848631e-10]
[105453320001.5807, 46046282215.53942]
[1.6772207337968798e-155, -7.745032801736509e-13, -1.0449780374860391e-13, 2.438379626417987e-11, -0.0028212433802508293,
5.387591937642218e-16, 4.510521514309953e-16, 1.9389130571725395e-05, -1.3686984102916085e-06, -8.885053688459976e-09,
5.302562303848631e-10]
[105668713779.58401, 45787776103.94903]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.0449780374860393e-13, 2.4383796264179863e-11, -0.00284615970945152З,
5.246438581531512e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.3686984102916077e-06, -8.885053688459974e-09, 5.302562303848631e-10]
[105516590259.86559, 45932073225.118484]
[1.6772207337968796e-155, -7.745032801736508e-13, -1.0449780374860391e-13, 2.438379626417987e-11, -0.0028305558160980297,
5.334836109815006e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.3686984102916081e-06, -8.885053688459976e-09, 5.302562303848631e-10]
[58512396553.23903, 52852336878.69969]
[1.4567110815902154e-155, -6.726768207800862e-13, -9.075913840978424e-14, 2.1177979447498638e-11, -0.0024601977778171778,
4.623339556799775e-16, 3.917508614845273e-16, 1.68399786605861e-05, -1.1887511890657059e-06, -7.716906849346833e-09, 4.6054172317275304e-10]
[95327384678.03574, 79359557772.13092]
[1.6772207337968796e-155, -7.745032801736507e-13, -1.0449780374860391e-13, 2.4383796264179866e-11, -0.002849550109869447,
5.227231643234366e-16, 4.510521514309953e-16, 1.93891305717254e-05, -1.347257454136562e-06, -8.885053688459976e-09, 5.302562303848631e-10]
[105551616090.82175, 45883165294.233765]
```



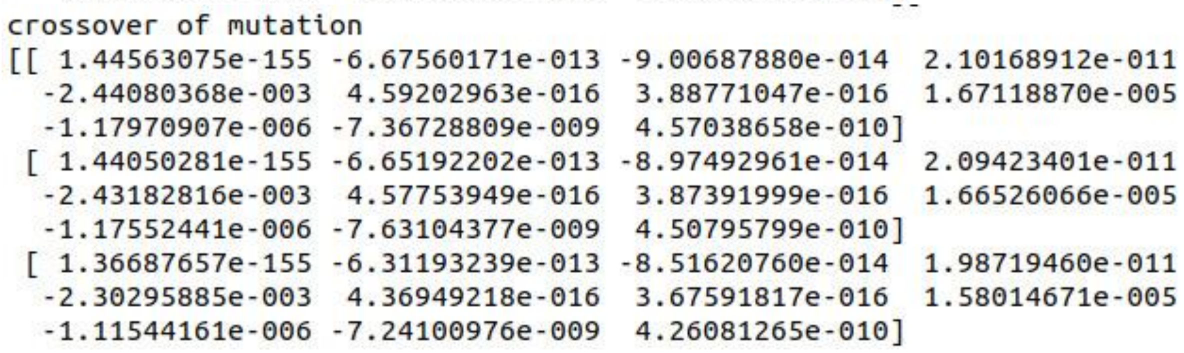
```
first crossover
[[ 1.51200227e-155 -6.98209062e-013 -9.42040084e-014  2.19818146e-011
  -2.55697491e-003  4.77957709e-016  4.06620227e-016  1.74791599e-005
  -1.23387164e-006 -8.00981114e-009  4.78022127e-010]
 [ 1.26203490e-155 -5.82779684e-013 -7.86300053e-014  1.83477352e-011
  -2.11945262e-003  4.07323874e-016  3.39396923e-016  1.45894687e-005
  -1.02988541e-006 -6.68561246e-009  3.98994512e-010]
 [ 1.65359926e-155 -7.63595409e-013 -1.03026088e-013  2.40403822e-011
  -2.80481463e-003  5.17969086e-016  4.44699669e-016  1.91160599e-005
  -1.34942207e-006 -8.75991927e-009  5.22788261e-010]
 [ 1.48973101e-155 -6.87924692e-013 -9.28164171e-014  2.16580303e-011
  -2.51799315e-003  4.71664471e-016  4.00630856e-016  1.72216981e-005
  -1.21569715e-006 -7.89182949e-009  4.70981032e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 1.64420080e-155 -7.59255409e-013 -1.02440525e-013  2.39037454e-011
  -2.78836434e-003  5.15313342e-016  4.42172156e-016  1.90074111e-005
  -1.34175245e-006 -8.71013105e-009  5.19816922e-010]
 [ 1.48033255e-155 -6.83584692e-013 -9.22308540e-014  2.15213935e-011
  -2.50154286e-003  4.69008728e-016  3.98103344e-016  1.71130493e-005
  -1.20802753e-006 -7.84204127e-009  4.68009692e-010]
 [ 1.87189692e-155 -8.64400417e-013 -1.16626937e-013  2.72140405e-011
  -3.18690486e-003  5.79653940e-016  5.03406090e-016  2.16396405e-005
  -1.52756419e-006 -9.91634808e-009  5.91803442e-010]
 [ 1.62192955e-155 -7.48971039e-013 -1.01052934e-013  2.35799611e-011
  -2.74938257e-003  5.09020105e-016  4.36182786e-016  1.87499493e-005
  -1.32357796e-006 -8.59214940e-009  5.12775827e-010]]
```



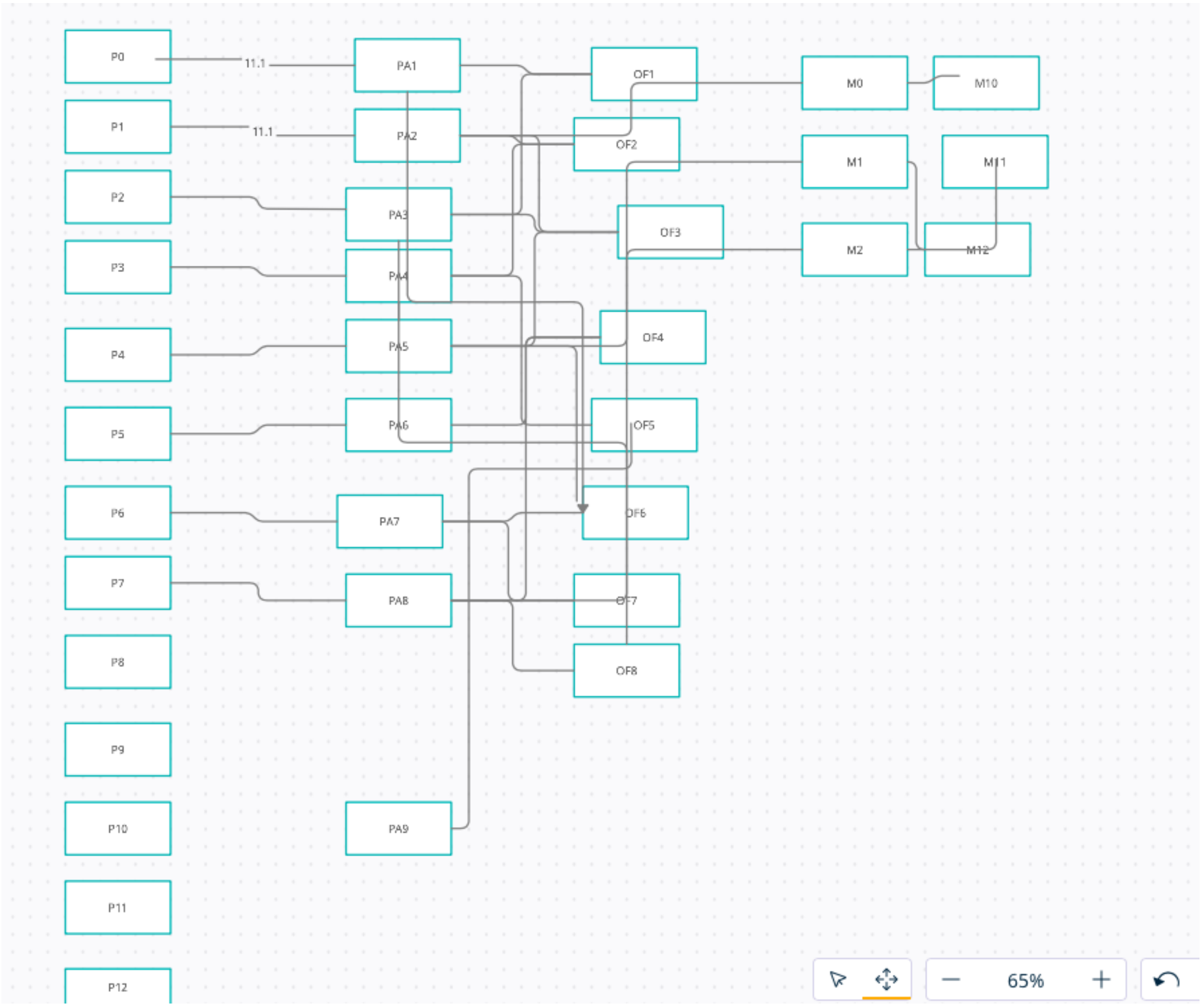
```
double crossover
[[ 1.50881079e-155 -6.96735308e-013 -9.40051661e-014  2.19354162e-011
  -2.55138881e-003  4.77055886e-016  4.05761948e-016  1.74422655e-005
  -1.23126723e-006 -7.99290433e-009  4.77013135e-010]
 [ 1.64454713e-155 -7.59415338e-013 -1.02462103e-013  2.39087804e-011
  -2.78897053e-003  5.15411207e-016  4.42265295e-016  1.90114148e-005
  -1.34203507e-006 -8.71196575e-009  5.19926416e-010]
 [ 1.51508061e-155 -6.99630572e-013 -9.43958018e-014  2.20265682e-011
  -2.56236299e-003  4.78827562e-016  4.07448080e-016  1.75147464e-005
  -1.23638373e-006 -8.02611861e-009  4.78995350e-010]
 [ 2.28936701e-155 -1.05717883e-012 -1.42637054e-013  3.32833107e-011
  -3.91761013e-003  6.97619392e-016  6.15675620e-016  2.64657090e-005
  -1.86824127e-006 -1.21278901e-008  7.23787333e-010]
 [ 0.00000000e+000  0.00000000e+000  0.00000000e+000  0.00000000e+000
   2.56935950e-289  0.00000000e+000  0.00000000e+000  0.00000000e+000
   0.00000000e+000  0.00000000e+000  0.00000000e+000]
 [ 8.44564805e-156 -3.90001267e-013 -5.26198880e-014  1.22784650e-011
  -1.38874736e-003  2.89358422e-016  2.27127393e-016  9.76340021e-006
  -6.89208333e-007 -4.47407040e-009  2.67010621e-010]
 [ 1.48665267e-155 -6.86503183e-013 -9.26246237e-014  2.16132768e-011
  -2.51260507e-003  4.70794618e-016  3.99803003e-016  1.71861117e-005
  -1.21318507e-006 -7.87552202e-009  4.70007809e-010]
 [ 1.62158321e-155 -7.48811110e-013 -1.01031356e-013  2.35749260e-011
  -2.74877638e-003  5.08922241e-016  4.36089647e-016  1.87459456e-005
  -1.32329533e-006 -8.59031470e-009  5.12666333e-010]
 [ 1.49292249e-155 -6.89398446e-013 -9.30152595e-014  2.17044288e-011
  -2.52357925e-003  4.72566295e-016  4.01489136e-016  1.72585925e-005
  -1.21830156e-006 -7.90873630e-009  4.71990024e-010]]
```



```
first mutation
[[ 1.45671108e-155 -6.72676821e-013 -9.07591384e-014  2.11779794e-011
  -2.46019778e-003  4.62333956e-016  3.91750861e-016  1.68399787e-005
  -1.18875119e-006 -7.71690685e-009  4.56236519e-010]
 [ 1.45671108e-155 -6.72676821e-013 -9.07591384e-014  2.11779794e-011
  -2.46019778e-003  4.62333956e-016  3.91750861e-016  1.68399787e-005
  -1.18875119e-006 -7.43990497e-009  4.60541723e-010]
 [ 1.67722073e-155 -7.74503280e-013 -1.04497804e-013  2.43837963e-011
  -2.83518320e-003  5.30862153e-016  4.70299501e-016  1.93891306e-005
  -1.36300522e-006 -8.88505369e-009  5.30256230e-010]
```



```
crossover of mutation
[[ 1.44563075e-155 -6.67560171e-013 -9.00687880e-014  2.10168912e-011
  -2.44080368e-003  4.59202963e-016  3.88771047e-016  1.67118870e-005
  -1.17970907e-006 -7.36728809e-009  4.57038658e-010]
 [ 1.44050281e-155 -6.65192202e-013 -8.97492961e-014  2.09423401e-011
  -2.43182816e-003  4.57753949e-016  3.87391999e-016  1.66526066e-005
  -1.17552441e-006 -7.63104377e-009  4.50795799e-010]
 [ 1.36687657e-155 -6.31193239e-013 -8.51620760e-014  1.98719460e-011
  -2.30295885e-003  4.36949218e-016  3.67591817e-016  1.58014671e-005
  -1.11544161e-006 -7.24100976e-009  4.26081265e-010]
```



## Crossover Function:

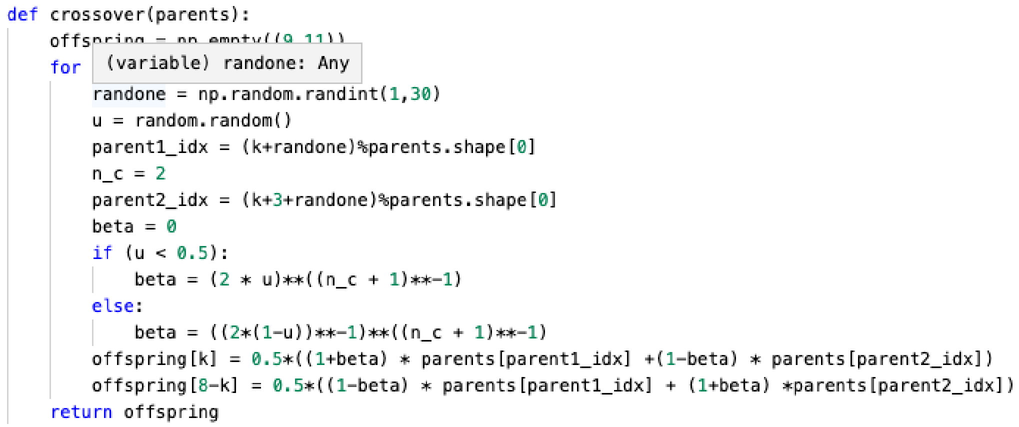
- The crossover function is done so as to get the children from the parents to get a offspring.
- We did crossover in a random and we got children By crossing vectors form the parent population.
- The intial crossover is not simulated binary crossover it is a simple crossover of parents.
- We did simulated crossover after a series of a good number of normal crossover we found this method when are searching how to get best vector
- The idea of this is simple as below:
-

$$\frac{x_1^{\text{new}} + x_2^{\text{new}}}{2} = \frac{x_1 + x_2}{2}$$

-
- Now this converts to more simpler way as follows:

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \leq 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases}$$

- Now we implemented as below.
- We have selected parents randomly as parent1_idx and paren2_idx and crossed then to get children.



```
def crossover(parents):
    offspring = np.empty((9,11))
    for  (variable) randone: Any
        randone = np.random.randint(1,30)
        u = random.random()
        parent1_idx = (k+randone)%parents.shape[0]
        n_c = 2
        parent2_idx = (k+3+randone)%parents.shape[0]
        beta = 0
        if (u < 0.5):
            beta = (2 * u)**((n_c + 1)**-1)
        else:
            beta = ((2*(1-u))**-1)**((n_c + 1)**-1)
        offspring[k] = 0.5*((1+beta) * parents[parent1_idx] +(1-beta) * parents[parent2_idx])
        offspring[8-k] = 0.5*((1-beta) * parents[parent1_idx] + (1+beta) *parents[parent2_idx])
    return offspring
```

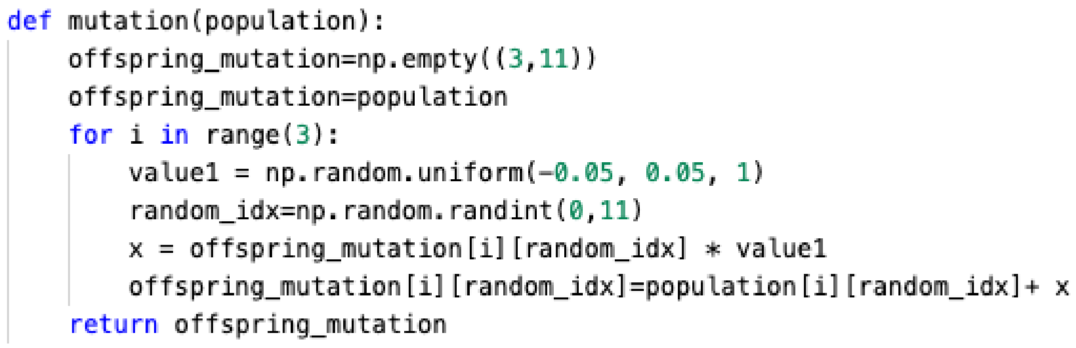
# MUTATION FUNCTION:

The mutation function is done as below and then it returns the offspring_mutaion.We decrease the size of the mutation from 6 to 3 as mutation vectors where too far away.

- We have done mutation with varying parameters in our process to get best vectors.

- We have selected some different numbers between different ranges and multiplied with the value in the vector which we wanted to change so that mutation is done. The  part of the vector is also selected randomly as  to maintain uniformity and not selecting same thing everytime.
- The range in which the vector is mutated is very crucial and we changed it between very small to small values.
- As a small change can effect the vector and it’s errors in a negative way which we found so many times we had to be more careful everytime we did it.
- The mutation population may seem small but it was changed in the process.

  The mutation is done as shown below



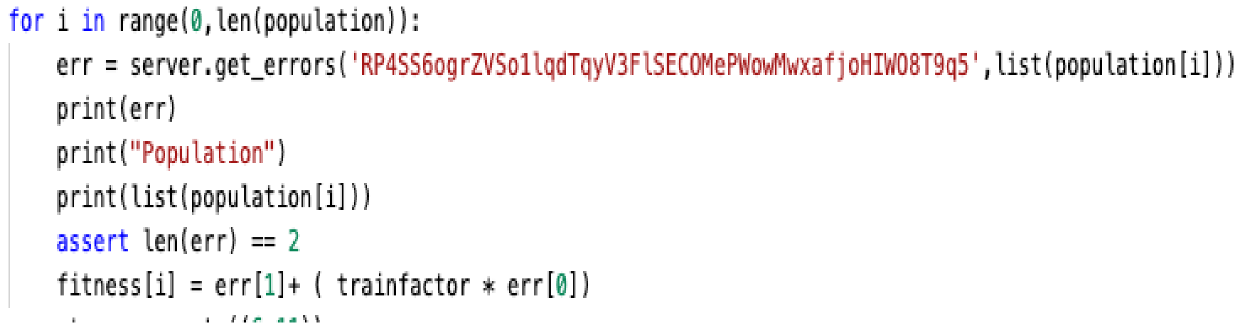
```
def mutation(population):
    offspring_mutation=np.empty((3,11))
    offspring_mutation=population
    for i in range(3):
        value1 = np.random.uniform(-0.05, 0.05, 1)
        random_idx=np.random.randint(0,11)
        x = offspring_mutation[i][random_idx] * value1
        offspring_mutation[i][random_idx]=population[i][random_idx]+ x
    return offspring_mutation
```

## FITNESS FUNCTION :

For the created population then we calculated the fitness of the each vector by getting errors from the get_errors of each vector.

- We have calculated errors for each vector in population and then took parents from them as we told above.
- Better the fitness Better the vector.
- We got two vectors from the get_errors and I used a combination of both to calculate fitness.
- From fitness I have assigned parents for next generation.
- Fitness is calculated as err[i] + (trainfactor * error[0])
- Train factor is changed from time to time.By which we obtained a balance between the validation and train error where in intial given overfit vector there is a large gap.
- Intially the train error is low and validation error is high so we put a factor of 0.x to give more importance to validation error we increased values and at 0.7 not exactly we found a balance between these two.

The code for calculating the fitness is shown below.



```
for i in range(0,len(population)):
    err = server.get_errors('RP4SS6ogrZVSo1lqdTqyV3FlSECOMePWowMwxafjoHIWO8T9q5',list(population[i]))
    print(err)
    print("Population")
    print(list(population[i]))
    assert len(err) == 2
    fitness[i] = err[1]+ ( trainfactor * err[0])
```

# HYPER PARAMETERS:

- POPULATION SIZE: We initially had the population around 40 then we changed to 24 so as to reduce the size because we have small number of requests per day and to use them effectively we did this.
- CROSSOVER POPULATION COUNT: Intially we distributed the population uniformly across the mutation and crossover as 6

but as we proceeded we we have increased the crossover the population to 9 and the double crossover to 9 so total 18.

- MUTATION POPULATION: We initially distributed population uniformly between mutation and crossover because initially we thought and then decreased the population to a minimal amount of 3 because they were similar and they impacted us in a negative way.
- TRAINFACTOR:We used train factor as to give more importance to test error than the validation one.First we considered only one but later we gave importance but less to test error.
- MUTATION RANGE:We had varied this from around a range of

  ( -1.x,+1.x) to a small range of (-.x,+.x) we have done this to keep our view sometimes large when we are getting similar vectors to a small value when vectors obtained variey considerably.

- DISTRIBUTION INDEX:we varied the distribution index between 2 to 5 according to what we do in simulated binary then what higher value of it indicates parents and children are closer to each other.

## HUERISTICS APPLIED:

- **TRAINFACTOR:**

  First we took only test error and proceeded but It produced better results only for some generation then they found not to decreases so I implemented this so as to get better at both of them and at this point errors were still high.

- **MODIFYING TRAIN FACOTR:**

  At start we have given a fixed train factor of 0.7 so as to give importance but less to validation error but after I discovered I neglected one and the are going far away I decided to change from time to time so as to get low values for both. This helped us but not in a great way.

- **SIMULATED CROSSOVER:**

  we implement this after long time of starting project where there was no improvement in vectors.

  in this also we changes the distribution index from time to time.

These are the heuristics applied

## STATISTICAL INFORMATION:

We have changed the train factor from 0 to 2.What I have observed is that is that when I had 0 then I observed that the difference grow between them and only one got better then I changes values between them so as to consider other one also when I did this I got better vectors I did’nt store these but this is what I have observed.