1. How do you describe the individuals and their chromosomes and how do you represent them in Python for the Genetic Algorithm method?

The individual is represented by a string that contains a 16 characters that represent each 4 a variable a0,a1,a2,a3, for bits for each variable and the chromosomes are represented in 4 bits, so that we can get a binary number in the range 0 to 15.

In python:

A individual: ‘0000000000000000’

Chromosome: ‘0000’

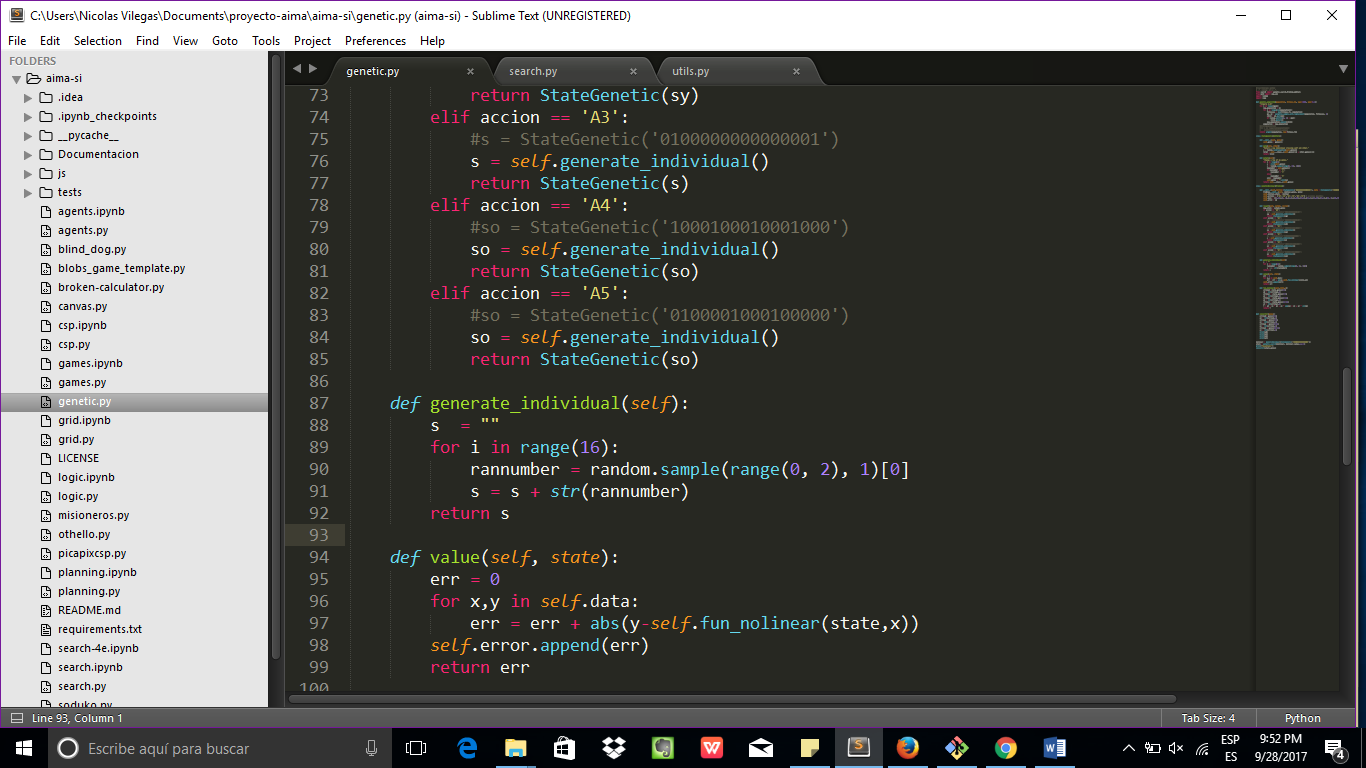
And then the chromosome can be turned in a value integer.

1. How do you generate the initial population?

At first, testing we generate a static population, 4 individuals at first then 6.

So we noticed that method is not effective to get a near solution compare with the real value, so we generate a random values between 0 to 1 to create an individual so that get a better solution to the real values.

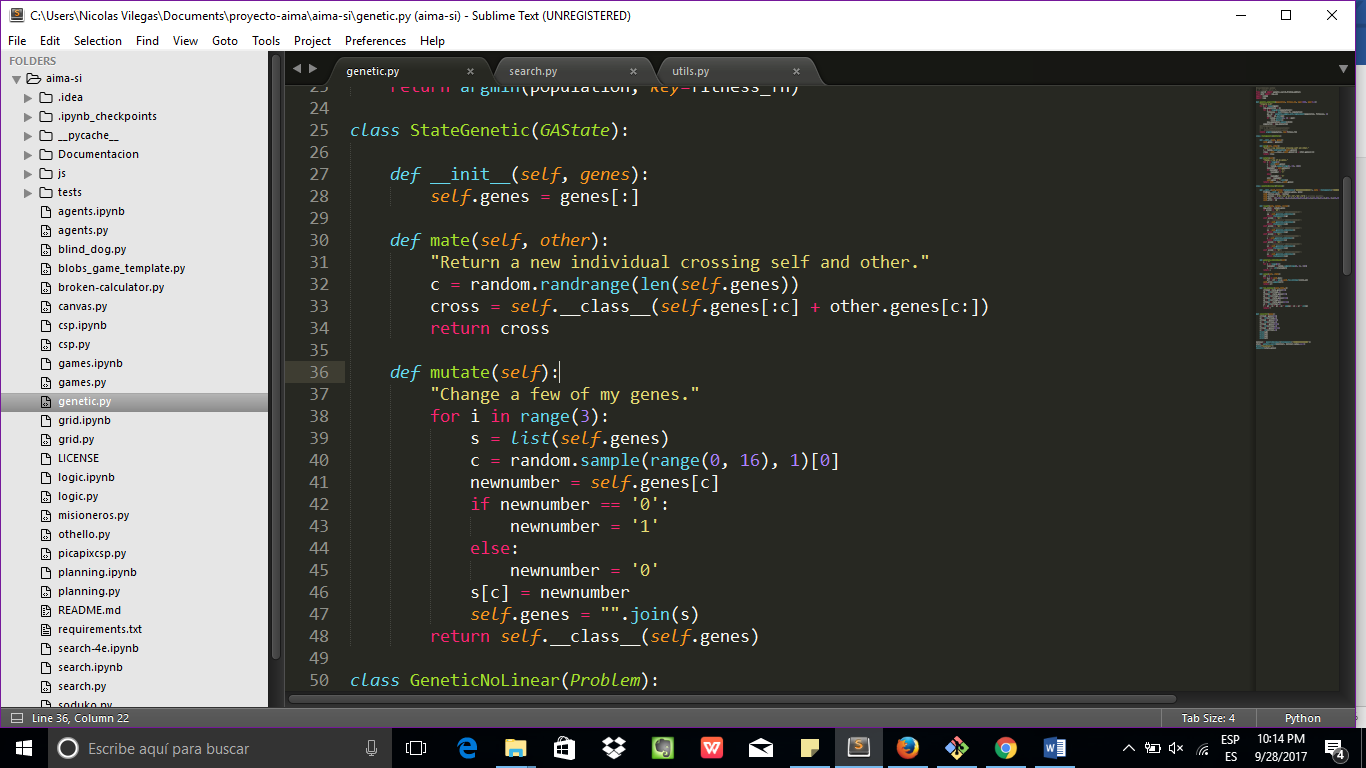
In the next code we show how generate an individual



1. How are the offspring of the current population generated during the execution of the method?

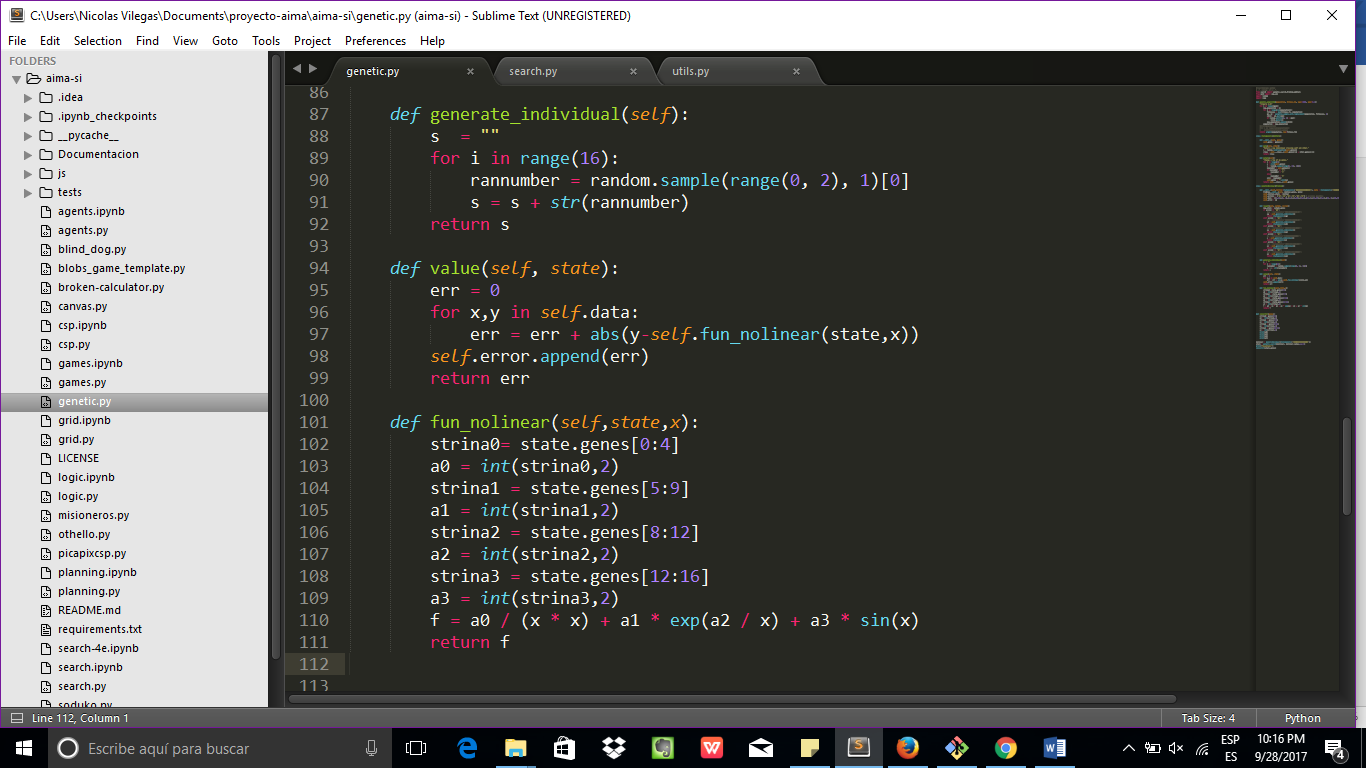
Well, firstly we generated a random number that indicates the size of the substring of the first parent to take, namely, we take the substring 0 to the number generated for the first parent and then take a substring by number generated plus one to end of the parent 2, and then we merge both substring and generate a new child.

In code:



1. How do you calculate the fitness of the individuals you want to optimize?

Well, to calculate a fitness we take the sum the absolute of the error with every one of the population, evaluating for each x in the formula with the new constants a0, a1, a2, a3 calculated in the population , to that we create a function named value that we use to replace the fitness function and get the value to get the most significative elements.



You must experiment with different population sizes, number of iterations and probabilities for applying the genetic operators to solve the problem. Show results of the experiments

Case 1:

With static population

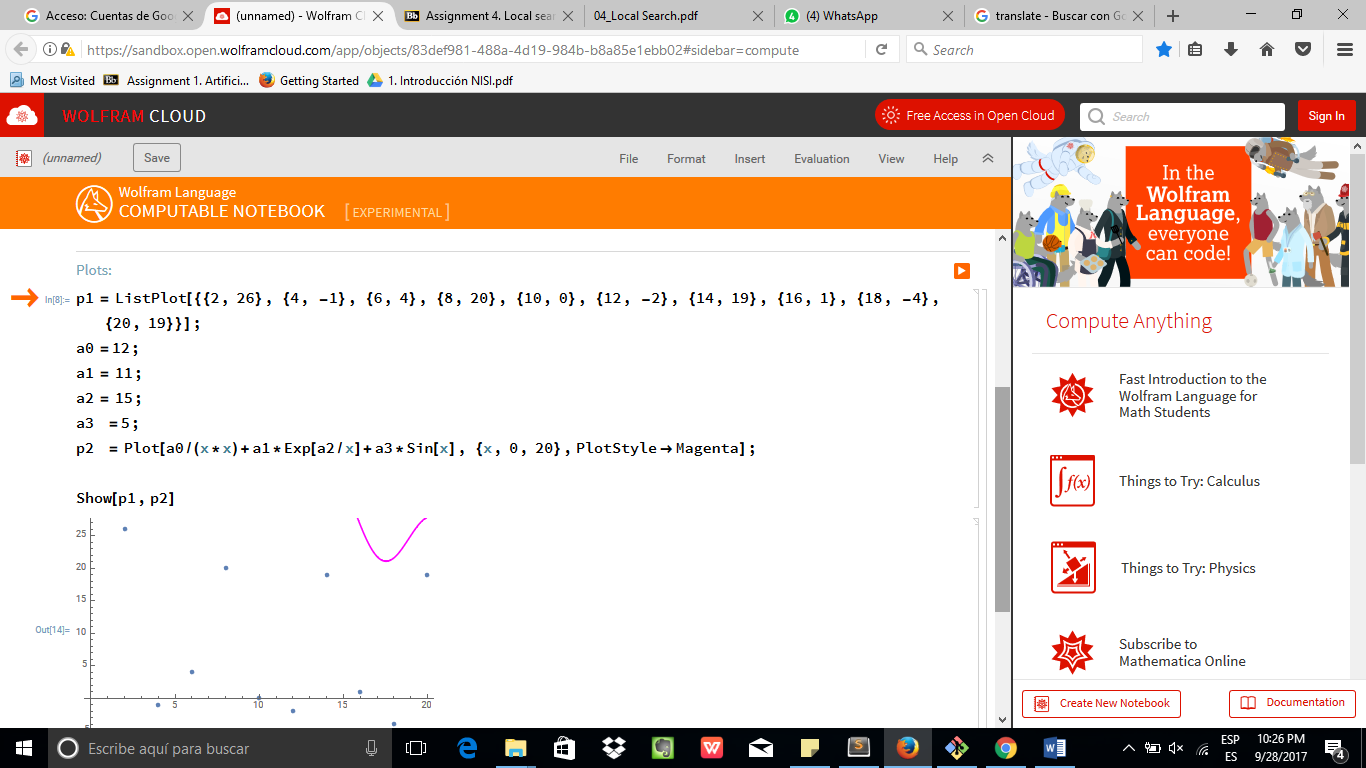
Individuals: 6

Mutation probability: 0.20

Number Generations: 1000

Result:

Its to away for the expected value.



Well first we decrease the numbers of the population and increase the generations with a static initial population:

Case 2:

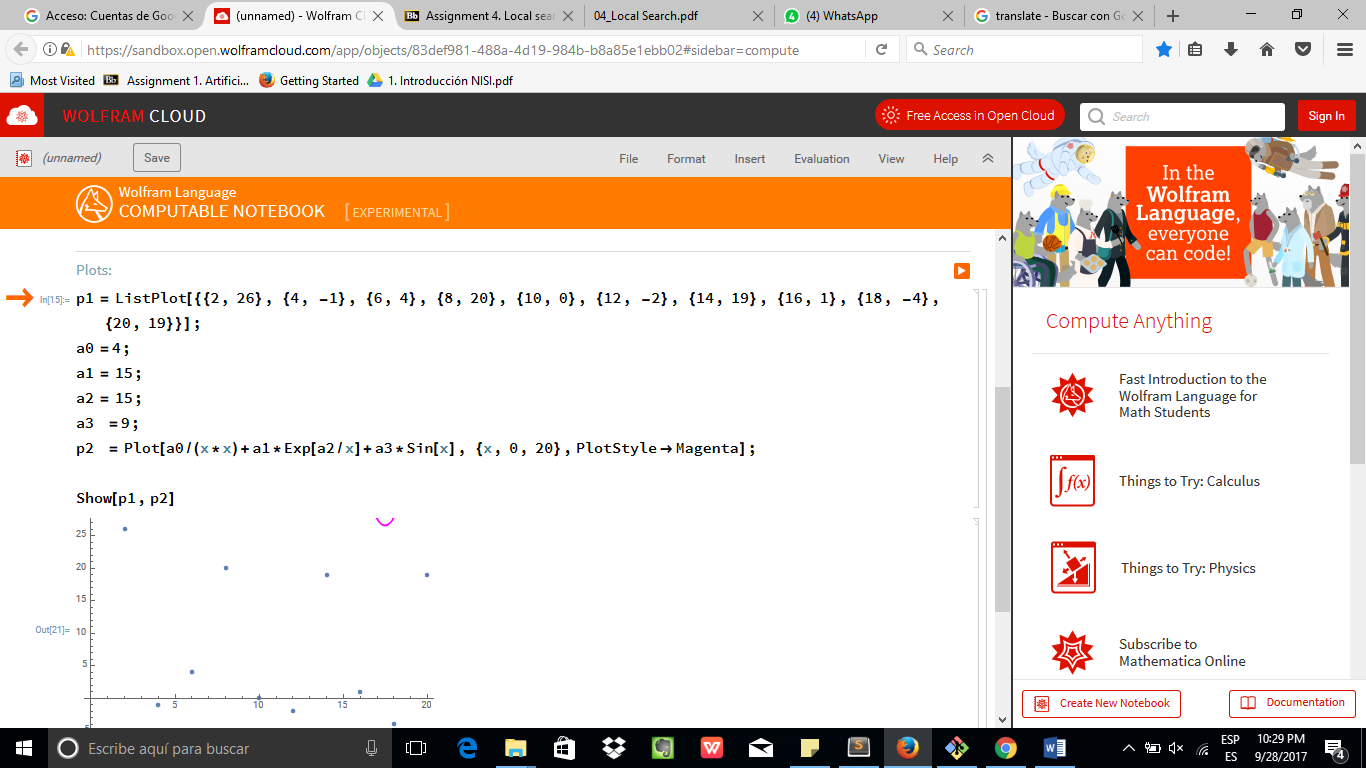
Individuals: 4

Mutation probability: 0.20

Number Generations: 2000

Result:

We get a worst estimation to the expected value:



So we change the probability to mutation and the number of generation with the same numbers of individuals

Case 3:

With random initial population

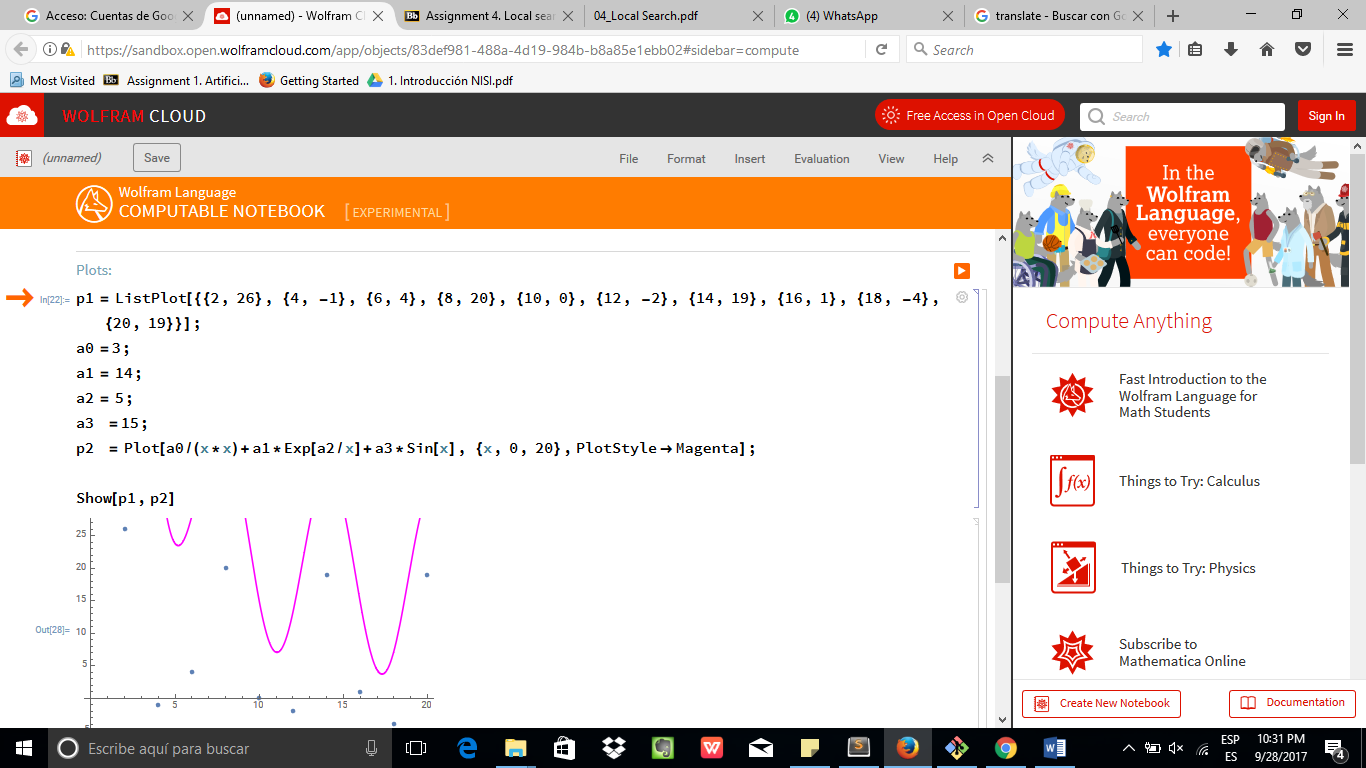
Individuals: 4

Mutation probability: 0.80

Number Generations: 1000

Result:

We get a better solution



Case 3:

So finally if we increase the numbers of the individuals

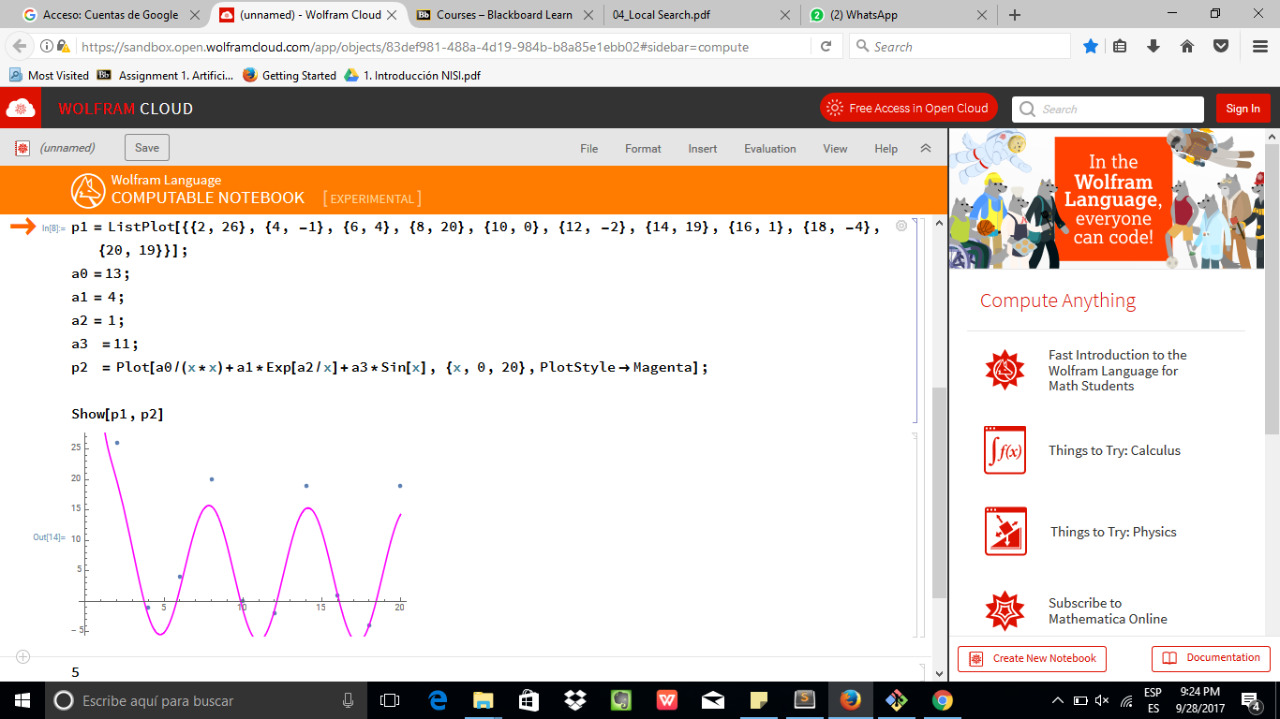
With random initial population

Individuals: 6

Mutation probability: 0.90

Number Generations: 1500

Result:



We get a better solution when the individuals are generated randomly and the probability of the mutation increase.