AI Lab 4 part 1

CoEvolution

Submitted BY :

Abed Abo Hussien 208517631

Samer Kharaoba 209050202

First of all how we implemented the network :

1. Create object: creates a network (couples of places [3,5] for example ) with up to 10% more comparators than the optimal configuration .
2. A network of 4 element would be : ([1,3],[2,4],[3,2],[1,4]) for example
3. Solve\_network: given a set of input check the output of the network
4. Check solution: checks if the given set is ordered correctly
5. Compare : takes 2 places in the network ,compares them and swaps them if there is a need to do that
6. Get depth, depth check, get depth at index: all work together to find the depth of the current network
7. Apply : applies a set of inputs on the network and by that calculates it’s fitness
8. Valid insertion : some insertions cannot be done such as [1,3],[1,3] in a row defeats the purpose of doing the first [1,3] comparison.

Text

Description automatically generated

Now that we explained what the network looks like we will now first explain section 1 in the report :

Implementation of the fitness function :

1. Go over all given sets to sort
2. Give each one a number of networks that was tested on it
3. Give each one the number of networks that successfully reordered the set
4. After all the sets have been checked get the percentage of failure denoted by :

* (1-sum\_of\_sets/len(sets))

1. Check the depth of that network and make it affect 50% of the fitness

Text

Description automatically generated

Implementation of fitness on the parasites :

The exact opposite of the networks fitness :



Compare:

Text

Description automatically generated

Both of those fitness functions are implemented in our fitness class , so that when we mutate one of them we can use those functions here:

Text

Description automatically generated

The algorithm returns a depth based output :

Text

Description automatically generated with medium confidence

And by the end of the algorithm displays a plot of the network such as : Chart, line chart

Description automatically generated

Output for input of size 6 : Text

Description automatically generated

-Changes done to the genetic algorithm :

1. It now holds 2 populations , sorted networks and parasites in our code we still call the parasites self.population
2. Here we initiate both populations :

Text

Description automatically generated

1. We used 500 sorted networks , generated randomly

- Changes done to the mutation operators :

1. Here we don’t select arbitrarily we get to know which are in the current,previous and next depth so that we don’t chose them :Text

   Description automatically generated
2. Modify operator : basically choses a comparator and stretches it or the opposite like : Text

   Description automatically generated

-Cross operation wasn’t changed , but we chose the single cross.

Section 2 :

Results :

On the 6 numbers we got the following results :

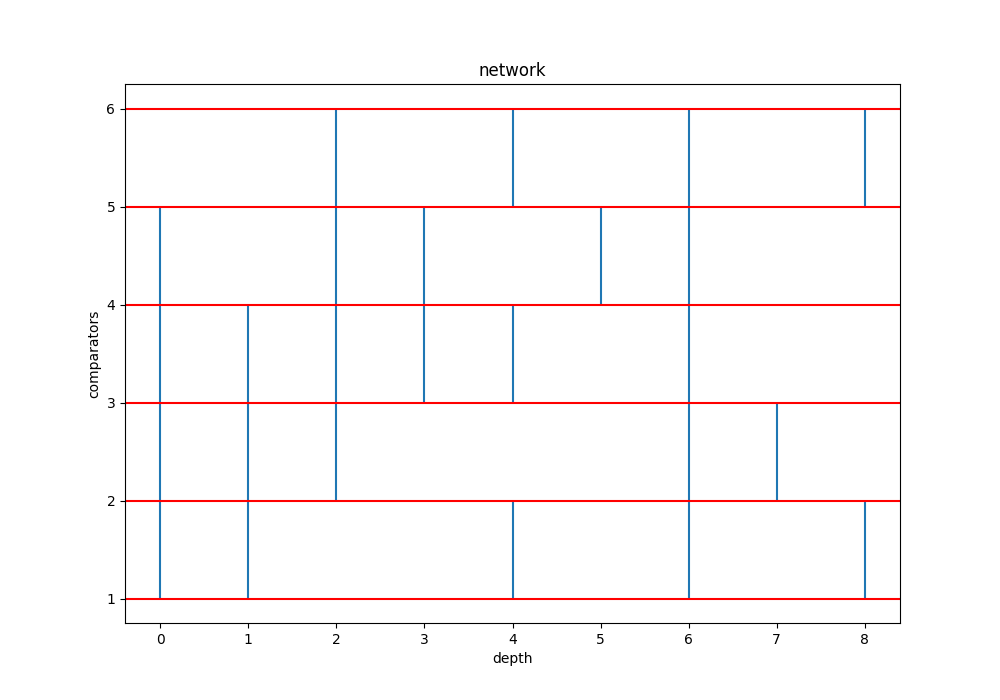
Note the the bar below isn’t depth , this was a typo :

Chart, histogram

Description automatically generatedGraphical user interface, chart

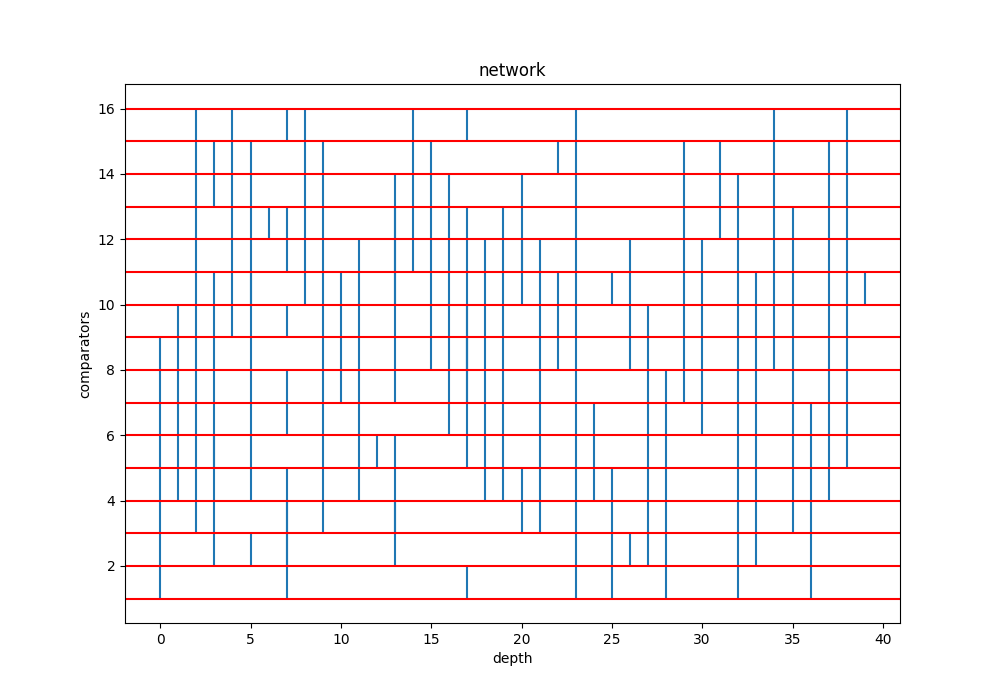
Description automatically generatedGiven one graph over 100 iterations :

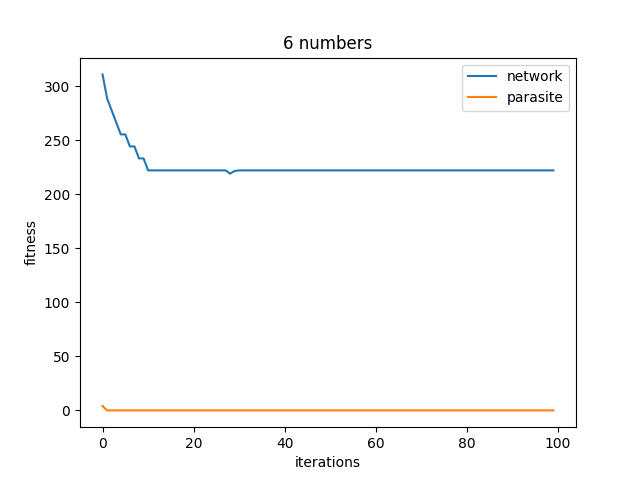
Result:



On the 16 numbers we got the following results :

Note the the bar below isn’t depth , this was a typo :

Given one graph over 100 iterations :  
these are for the 16 numbers :  
 we can clearly see that our algorithm didn’t succeed to get the network to a fitness of 0 because the solution that we got in the results is composed of 61 comparitors and isn’t the optimal one , although the parasites seem dormant , they aren’t but the algorithm falls to a local minima pretty quickly :

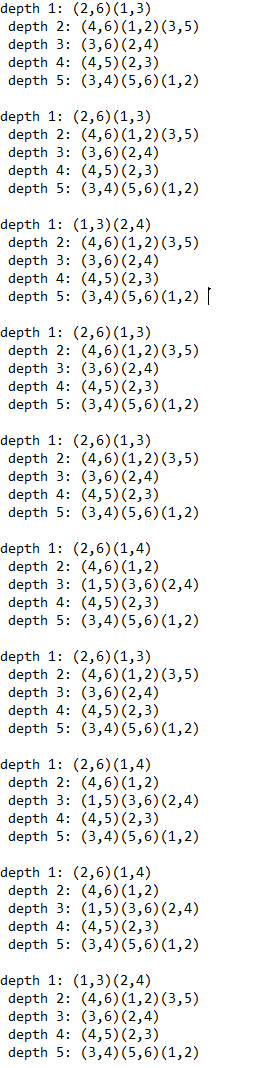


Section 3 :

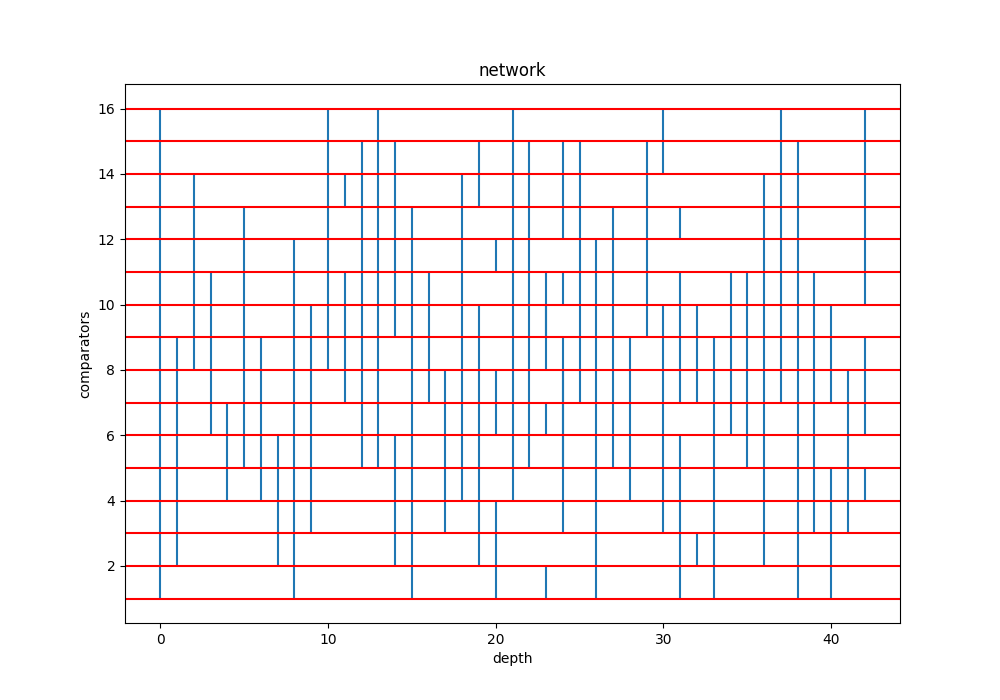
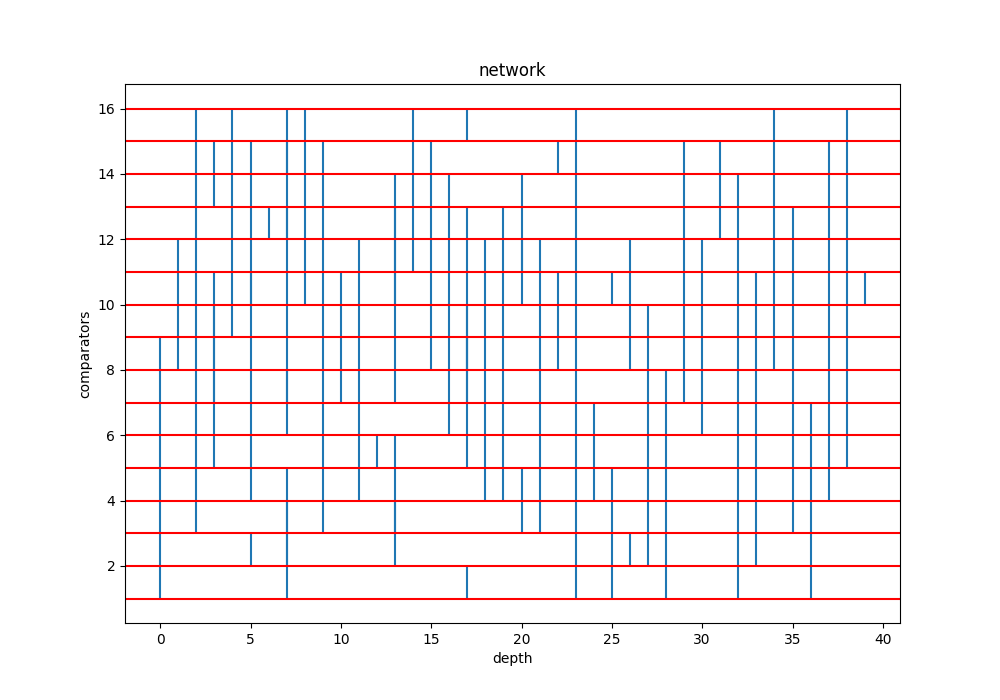
Text, letter

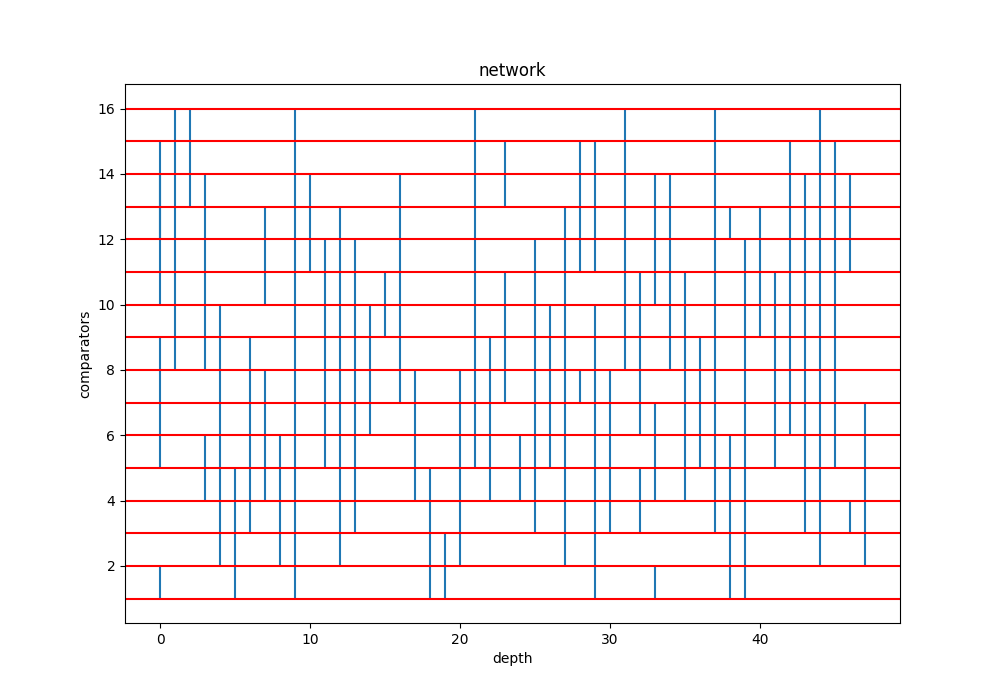
Description automatically generatedIn the following results we can see Strategy Recycling effect as the top 10 solutions are mostly recycled solutions mashed together at the top as they are the top performers :

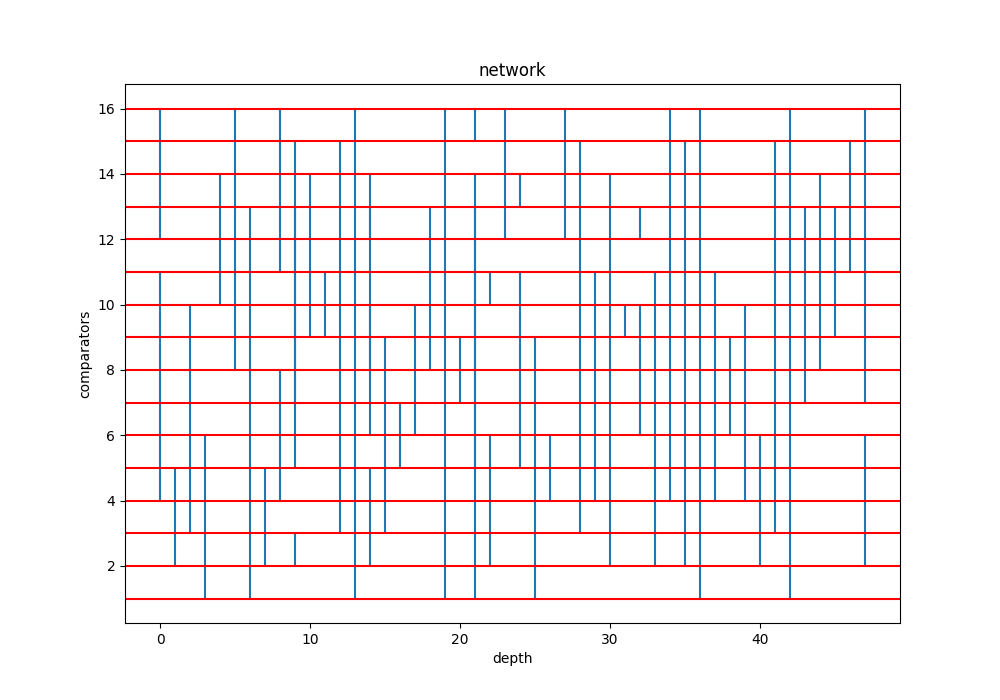
Text

Description automatically generated

As are the sorting networks :





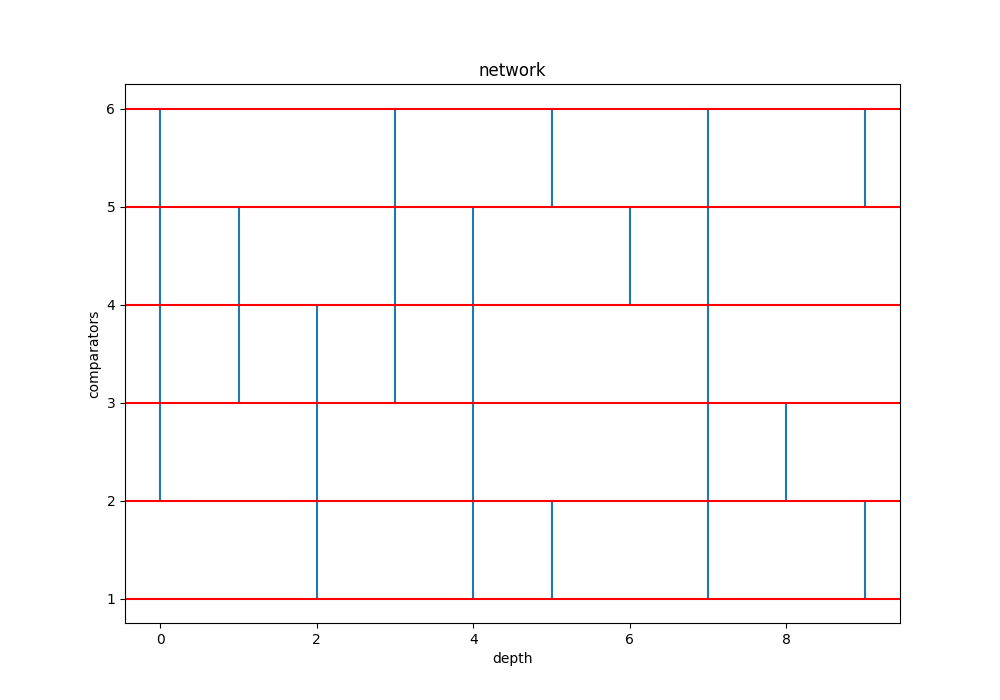


Chart

Description automatically generated

Chart

Description automatically generated



Chart

Description automatically generated

All output files that are not in a folder are for the top 10 performers in each problem set:

A picture containing calendar

Description automatically generated

Output format and demonstration of the depth dipping to the optimal :

Text

Description automatically generated

Conclusion:

1. We can improve on the algorithm farther by using a basic solution as a base for creating population
2. We can use a different model of the GA as we have used a type of “species” GA
3. The fitness function of the depth might need to be given less control over the fitness of the networks maybe a 20-30% affect on it would be better
4. The regret that we have in this assignment is that we put ourselves under a lot of pressure , we are amazed by the time management skills that we acquired in this endeavor .

Instruction on usage :

1. just like all old assignments ,all the outputs are generated by a script that can be chosen in the main menu
2. you are given two options either manual or automatic ,
3. press the correct number to choose the aforementioned mode

Lastly , all the output files are generated automatically in the output folder :

Includes: results\_X\_numbers : this folder has the output of the top 10 solutions for k=16,6

Thank you for reading , we hope that you enjoyed