Assignment 2

Problem 1: Map Coloring

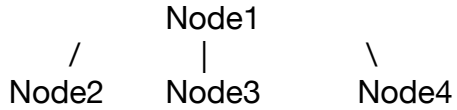
Studying the different algorithms implemented, we have a set of conclusions of each one and differences between them.

Algorithm Min-Conflicts: Min-Conflict uses a loop with maximum steps to find the solution, and in each step tries to find the node which has less neighbors with the same color as his own color. And then try to change this one until we find a feasible solution.

Algorithm Backtracking: Backtracking uses recursion to assign colors to each node, and goes back if it's not possible until it finds the solution in the tree. It works really faster than Min-Conflicts.

Algorithm Backtracking with forward checking: In the implementation of this algorithm, the only check achieved is for one step forward, but it makes a little improvement. The part of backtracking works almost the same as the previous one.

Example:



If in node1 the assign color is blue, then the other ones will not have that value in their domain.

Algorithm Backtracking with AC-3: This algorithm has the part AC-3 well implemented, but not good connected with the part of backtracking. The idea of this algorithm is to delete values of the domain that are not compatible with any constraint.

Comparison of efficiency of each algorithm:

To compare the efficiency of each implemented algorithm we have 4 tables that study the execution time in different problem parameters. First of all we have an execution with 10-coloring and then another with 50-coloring.

– – – – Min-Conflicts Algorithm
– – – – BackTracking
– – – – BackTracking with forward checking
– – – – BackTracking with AC-3

10-coloring
As we can see in the following table, backtracking with AC-3 is not working properly. The this is it's never deleting values because there's always possible values to fit the constraints and with the backtracking algorithm I go through all the nodes iterating in sequential id's instead of following the net of neighbors.

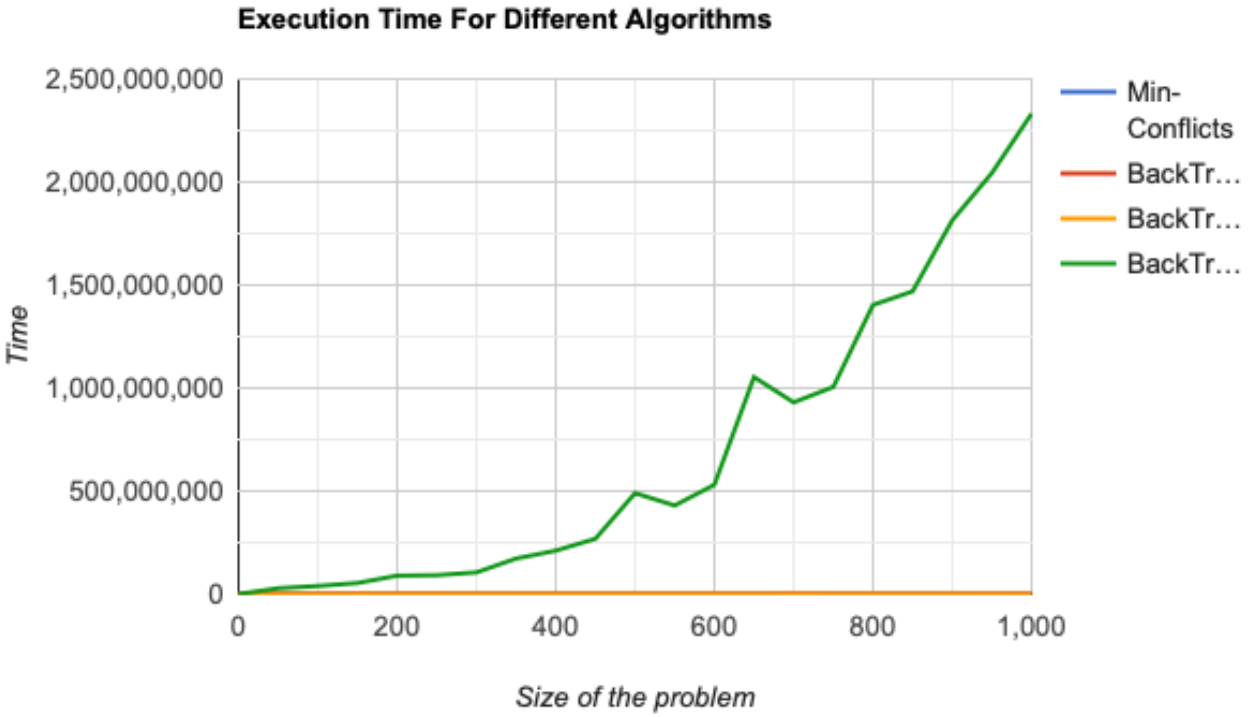


Figure 1: Line chart of_time executions of 4 different algorithms and using 10 colors

Now we omitted green line and we can focus in the other 3 algorithms. As we can see, Min-conflicts algorithm is not finding a solution with 10 colors if the size of the problem is bigger than 50. Probably if we increase the value of max-steps we would find it, but is obvious that when the size of the problem is 50 nodes, the Min-Conflicts algorithm is noticeably slower.

Moreover, comparing the other two backtracking algorithms we can observe that there's a tendency where the red line is above the orange one. Red line is using only backtracking and the orange is adding forward checking. It is clearly faster using forward checking, because we always omit one value in the next node.

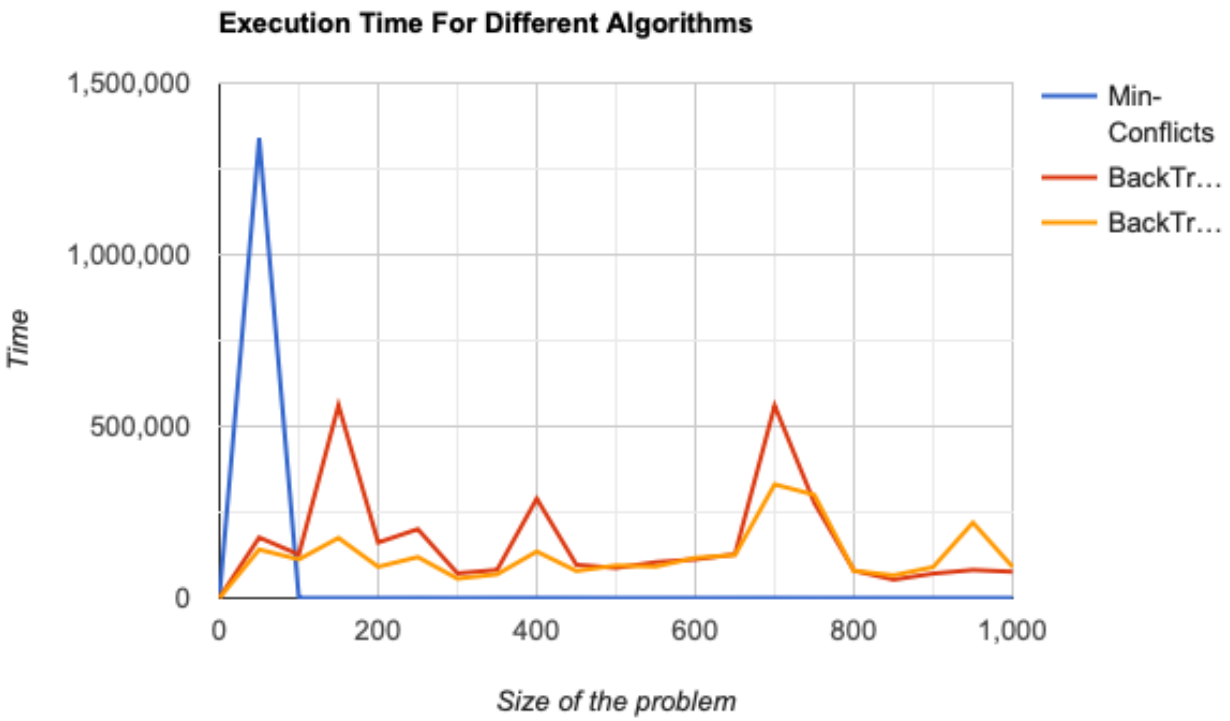


Figure 2: Line chart of_time executions of 3 different algorithms and using 10 colors

50-coloring

Then we have the same type of execution but using 50 colors to paint the nodes. Min-conflicts is obviously slower than the other two, but now is always finding a solution. To compare better the other two we will omit the blue line.

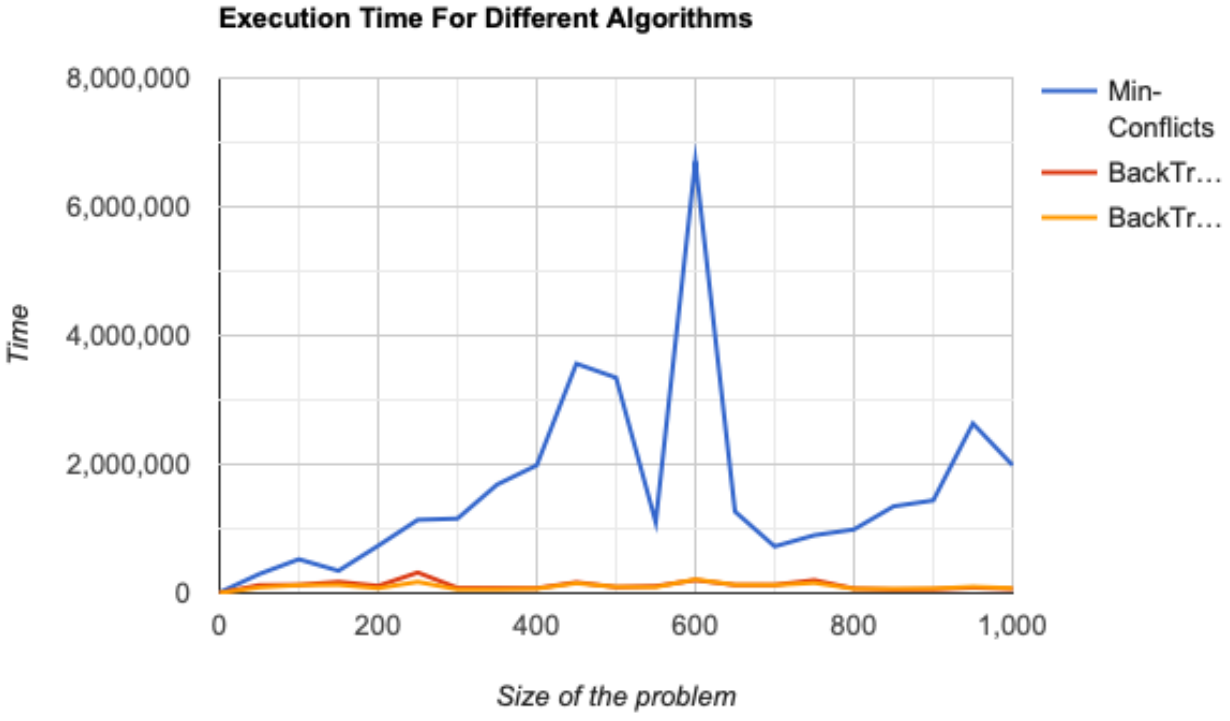


Figure 3: Line chart of_time executions of 3 different algorithms and using 50 colors

Comparing the other two lines we can see another time that backtracking with forward checking is faster than without it.

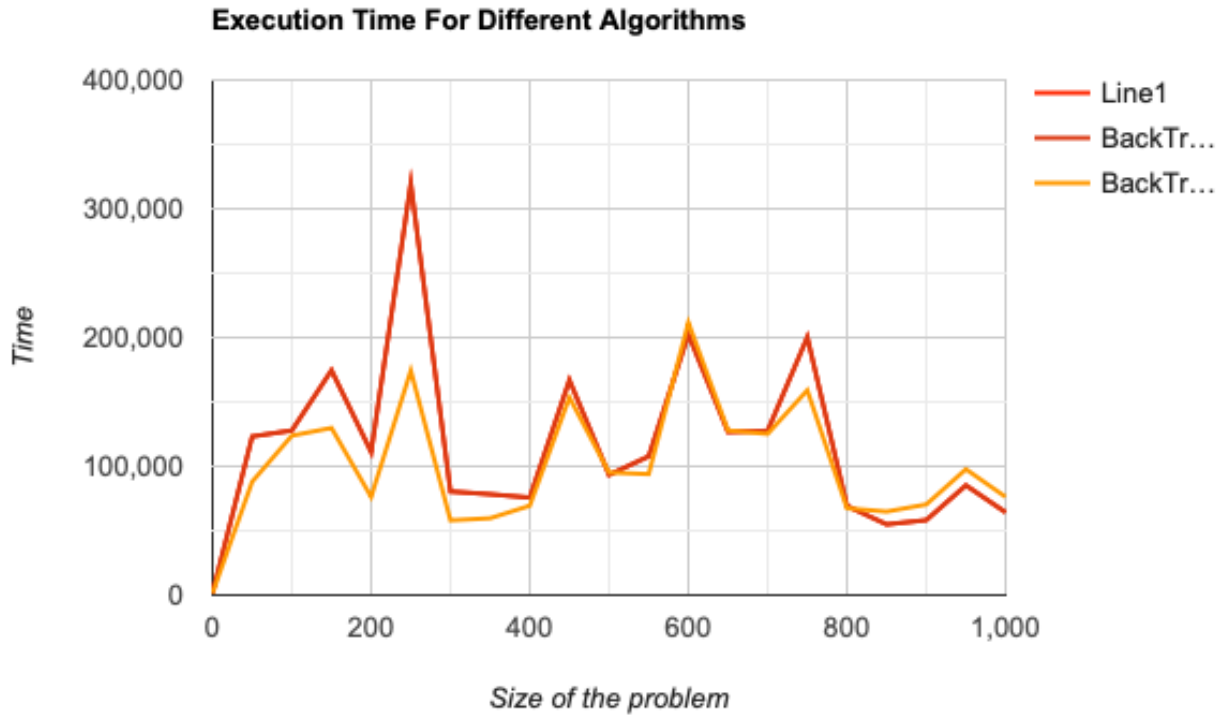


Figure 4: Line chart of_time executions of 2 different algorithms and using 50 colors

Problem 2: Zebra Puzzle

I really tried to write an AC-3 algorithm to solve Arc Consistency Problem but I couldn't and I got no solution. If I had more time I would try to do in another way but it's really late to try it.