Comparison Among the Performance of Arc Consistency Algorithms

Md. Musfiqur Rahman
Roll: 05

***Problem Statement*- Generate a constraint graph and show a comparison between the performance of AC-1, AC-2, AC-3, AC-4**

I. Problem Solution

We first construct a random graph. Each time Domain size is kept <=50 in range [0,1000] and used six constraints on each edge assigning randomly
. For node as X axis parameter :

$$Nodes : [3, 200], step = 5, Edges : [N, N * (N - 1)]$$

For edge as X axis parameter :

$$Edges : [3, 150], Step = 5, Nodes : [E + 1, 2 * E]$$

For graph density as X axis parameter :

$$Density : [0.01, 1], Step = 0.1, Nodes : [3, 200], Edges : d{*}N{*}(N{-}1)$$

Then run AC-1,AC-2,AC-3 and AC-4 on the same graph and used the average result for plotting the graph.

II. Performance Comparison

We observed execution time and number of constraints satisfied for understanding the performance.

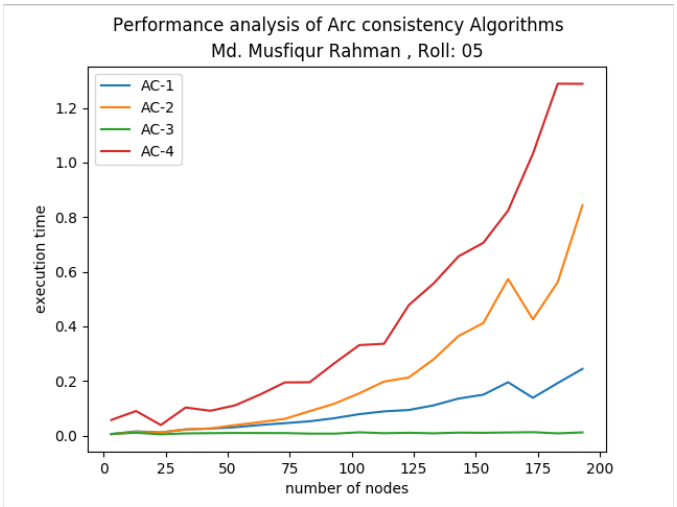

Fig. 1. Number of nodes vs execution time

1) Execution time increases with respect to number of nodes. AC-3 performs best and AC-4 performs worst. AC-1 is better than AC-2 here(Fig:1).
2) Execution time increases with respect to number of edges. Performance : AC3 >AC2 > AC1 > AC4 (Fig:2)

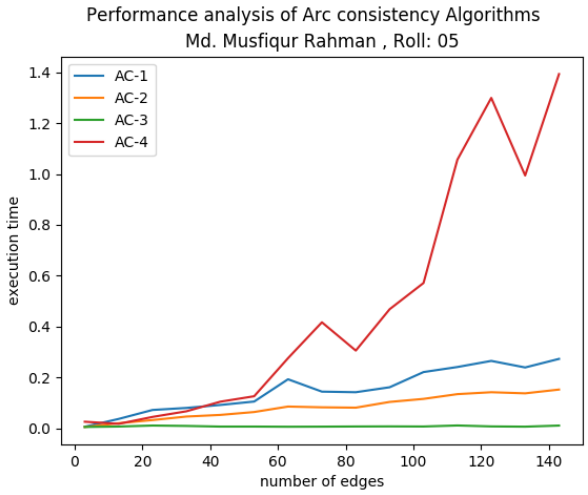

Fig. 2. Number of edges vs execution time

3) AC-4 acts worst in case of graph density as x axis parameter. Performance : AC3 >AC1 > AC2 > AC4 (Fig:3)

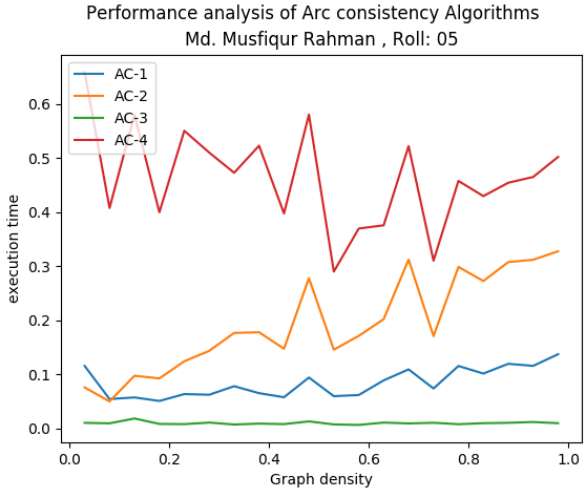

Fig. 3. Grpah density vs execution time

4) Number of nodes from . We observe AC-3 performs better for small number of nodes and AC-2 performs better for large number of nodes. This may change for different graphs. Number of constraint satisfied are from 0 to 500. (Fig:4)

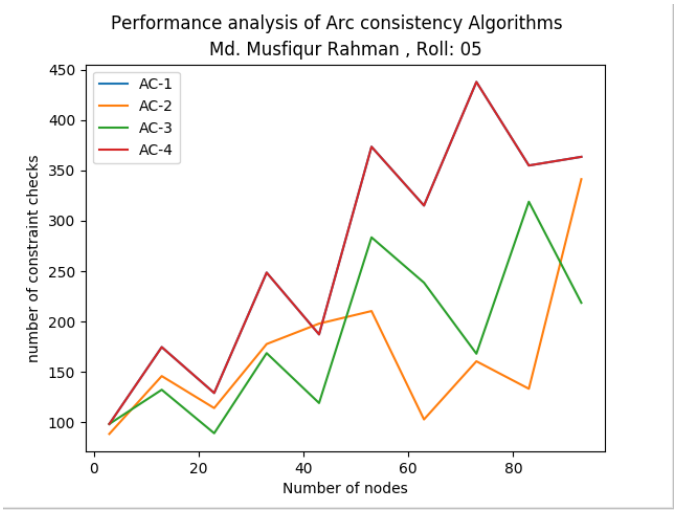

Fig. 4. Number of nodes vs number of constraint checks

5) Number of constraint checks increases with respect to number of edges. Performance : AC3 >AC2 > AC4 > AC1 (Fig:5)

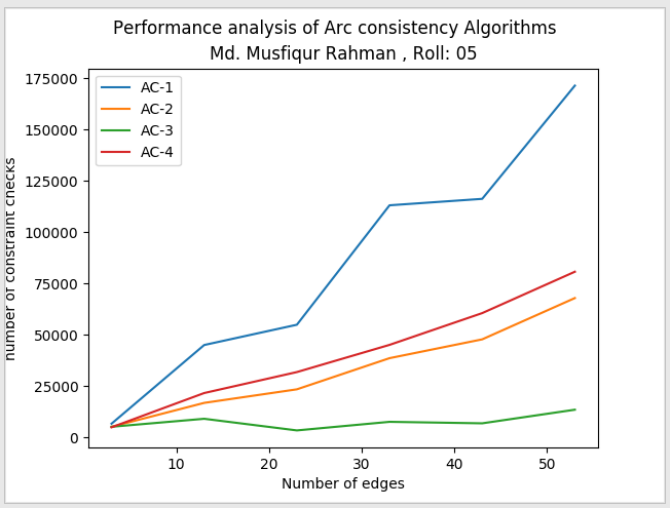

Fig. 5. Number of edges vs number of constraint checks

6) Number of constraint satisfaction decreases with respect to graph density.Because most of the domain values get reduced to satisfy the large number of constraints assigned to the edges. Performance : AC3 >AC2 > AC4 ≈ AC1 (Fig:5)

III. Correctness of Implementation

We can claim our implementation is correct since after each execution of the algorithms we get the same state of domains of each node. After plotting number of nodes vs domain reduction graph, we observe that all four lines combine together and form one single line which increases with respect to number of nodes.(Fig:7)

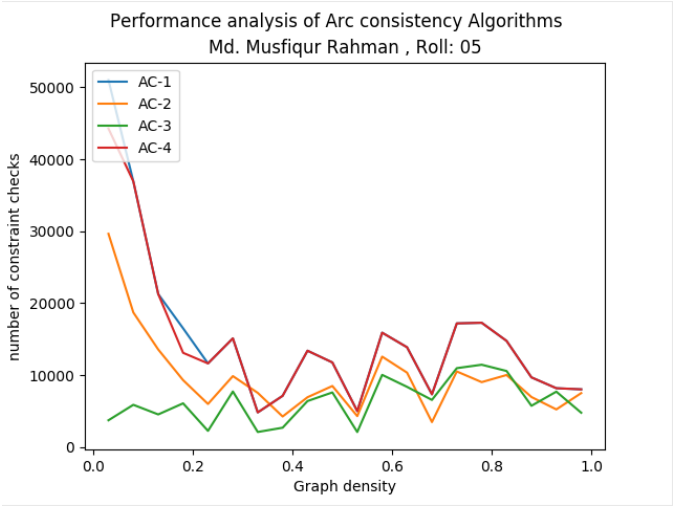

Fig. 6. Grpah density vs number of constraints

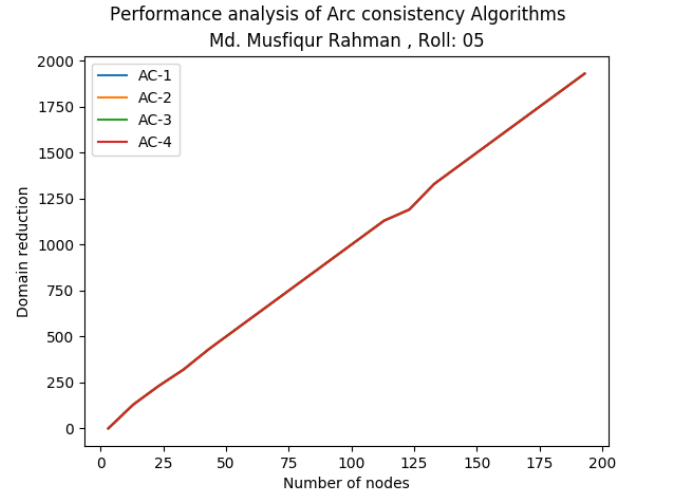

Fig. 7. Number of edges vs number of constraint checks

IV. Experiment Findings

In most cases, AC-3 performs better than AC-4. For AC-1 and AC-2 sometimes AC-1 acts better and sometimes AC-2. We performed the Anova Tukey HSD test and got each pairs of data significant.

treatments pair	Tukey HSD Q statistic	Tukey HSD p-value	Tukey HSD inferfence
AC-1 vs AC-2	43.8971	0.0010053	** p<0.01
AC-1 vs AC-3	12.7796	0.0010053	** p<0.01
AC-1 vs AC-4	7.3822	0.0010053	** p<0.01
AC-2 vs AC-3	56.6767	0.0010053	** p<0.01
AC-2 vs AC-4	36.5149	0.0010053	** p<0.01
AC-3 vs AC-4	20.1618	0.0010053	** p<0.01