Constraint Satisfaction problem (Solution Design)

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 definition

For designing a constraint graph , we can think of a situation where some agents(computer programs or suitable entity are communicating with each other to co-operate or compete with other agents to solve any problem. Here we will consider each node as an agent and the arc or edge between them as constraints which must be considered and satisfied.

II. Constraint specification

Constraints in a graph are used to achieve a goal by satisfying those constraints with allowed values in the domain.Now we will consider some specific constraint which are relatable with the situation mentioned above. If the values in the domain of the adjacent nodes can satisfy these constraint then they will not be reduced from the domain. We describe constraints in the following which will be used in our solution.Here X,Y will be chosen from first and second agent's domain in order. We will assign an id (like following) to each constraint and allocate an id to each edge randomly.

1) Agents starts communicating with each other if:
$$distance(X,Y) > gcd(X,Y) \tag{1}$$
2) Agent 1 will send data to agent 2 so that he can transfer data faster if data transfer rate of agent 2 and agent 1 is related by
$$Y = X^2 \tag{2}$$
3) Agent 1 and agent 2 will make a circle of radius 10 or greater with their domain values
$$X^2 + Y^2 > 10^2 \tag{3}$$
4) Communication between agent 1 and agent 2 will be stable if
$$Y = X \tag{4}$$
5) Agents co-operate with each other if the following equation has a real number solution. (b,c is from both agents and a=1)
$$ax^2 + b * x + c = 0 \tag{5}$$
6) Value of agent 1 and agent 2 will be allowed if:
$$X \oplus Y isOdd \tag{6}$$

III. Domain specification

D is a set of finite domains for the variables of the agents, with D_i being the domain of variable x_i of $agent_i$. We will generate our domain with two types of randomization:

1) Randomly choose the size of domain
2) Randomly choosing values for domain

Then we will assign these domains to random nodes. So that, when running our algorithms many values get reduced in the domain. We will mainly consider constraints for values of the domains.So,the i_{th} domain will be more suitable for i_{th} constraint(listed before).

1) D1:[50,90,130,200,240,390,...]
   -Considering distance between two agents.
2) D2: [1,4,9,16,25,36,...]
   -Perfect square values should occur in the domain.
3) D3: [9,10,11,12,13,14,...]
   -Since squared sum have to greater than 100
4) D4: [200,220,240,260,280,300,...]
   -Values of both agent have to be same so we choose larger values.
5) D5:[5,9,16,20,36,40,45....]
   -Constants of the equation will be taken from this domain of both agents.
6) D6:[29,39,49,59,69,79,...]
   -Domain values will have xor equal to an odd number.

IV. Program Execution

After we assign domain and constraints in the graph. We will do the following for n=10,,50,100,150,200 nodes

(a) We will run AC-I, AC-II, AC-III, AC-IV algorithms to get individual results and save their run time , number of edges considered in total and reduction percentage.
(b) We will do the same for 10-20 times for different graph with same number of nodes for getting the average value.

V. Result and Observation

After our program finished running, we will get a table of data which will indicate individual performance of our algorithms. To get a better visualization we will plot different graphs with number of nodes in X-axis and run time or number of edges considered in total or reduction percentage in Y-axis. We will get 4 lines in same graph each for four different algorithms.