

Constraint Satisfaction Problem Description

1 Problem statement

Comparison between AC 1, AC 2, AC 3 and AC 4 arc consistency algorithm in Constraint Satisfaction Problem.

2 Input

A Constraint Satisfaction Problem can be defined by a tuple $\langle X, D, C \rangle$. Here,

- X is a set of variables, $\{X_1, \dots, X_n\}$.
- D is a set of domains, $\{D_1, \dots, D_n\}$, one for each variable.
- C is a set of constraints that specify allowable combinations of values.

It is easy to visualize a CSP as a constraint graph. Constraint graph of a CSP is the graph whose nodes are the variables of the problem and an arc joins a pair of variables if the two variables occur together in a constraint.

2.1 Graph generation

We will generate a constraint graph with n nodes ($20 \leq n \leq 100$) randomly and also assign edges randomly between vertices. For graph generation, we will use the Erdős-Rényi model. It is simple and allows to control graph density.

2.2 Constraints

Each constraint lists all combinations of valid domain values for the variables participating in the constraint. A binary constraint relates two variables. For example, $\langle(X, Y), X \neq Y\rangle$. For our problem, we will use mathematical functions as binary constraints. We will assign one constraint randomly from the list of predefined constraints to every arc in constraint graph. Constraint lists are:

- $X > Y$
- $2 * X < 3 * Y$
- $Y = X^2$
- $X|Y$
- $gcd(X, Y) = 1$

2.3 Domains

For each variable, its domain consists of a set of allowable values, $\{v_1, \dots, v_k\}$ for this variable. Allowable values can be any number, letters, strings etc. However, our domains will consist of only numbers in this problem as our constraints are mathematical functions. For each node, we will select domain size randomly and assign random values from predefined range [1, 500]. There will be a minimum domain size aslo.

3 Implementation and comparison

After generating the constraint graph, we will run AC 1, AC 2, AC 3 and AC 4 arc consistency algorithms on the graph and record different performance metrics as instructed. Example metrics are running time, number of times a constraint is accessed etc.

For comparison, we will construct number of nodes vs running time for each algorithms. We can compare them from the output graph.