Constraint Programming
Quiz #04 (arc consistency)

Why is it sometime better not to enforce node consistency?

Why does the notion of arc consistency contain the word “arc”?

What is the major inefficiency of the AC-1 algorithm?

What is the major inefficiency of the AC-3 algorithm and how to remove it?

What is the worst-case time complexity of various AC algorithms (AC-1, AC-3, AC-4)?

What is the optimal worst-case time complexity of arc consistency?

Why is AC-4 used less frequently in practice?

Can we improve time efficiency of AC-3?

What is the major difference between AC-3.1 and AC-2001 algorithms?

Assume that variables in a CSP are ordered somehow. If we make the problem DAC in that order and then make the problem DAC in the reverse order, will we always get full AC?

Assume that variables in a CSP are ordered somehow. If the problem is DAC in that order and also in the reverse order, is the problem full AC?

Are there any CSPs where DAC always implies AC?

If a CSP is arc consistent, does that CSP (always) has a solution?

In which situations we know that an arc-consistent CSP has/hasn’t a solution?