Graph coloring problem
Sudoku Game

NO Date

Constraint Satisfaction Problem (CSP)

→ CSP consists of three components
V, D, C

→ V is the set of variable
$\{V_1, V_2, \ldots V_n\}$

→ D is the set of Domains one for each variable
$\{D_1, D_2, \ldots D_n\}$

⇒ C is the set of constraints that allowable combination of values
$C_j = (\text{scope}, \text{relation})$ $\{C_1 C_2 C_3\}$

where
↳ scope is the set of variables that participate in the constraint
↳ Relation is the relation that defines the values that variable can take

$C_1 = (V_1, V_2)$ ← scope

V_1	V_2
A	B
(1,2)	(2,4)

rel:
$V_1 \neq V_2$

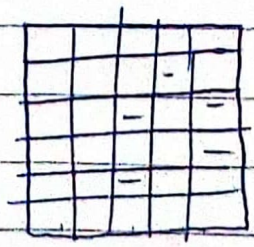
$C_1 = ((V_1, V_2), (V_1 \neq V_2))$

$C_1 = ((V_1, V_2), (1,2), (1,4), (2,4))$

Summary

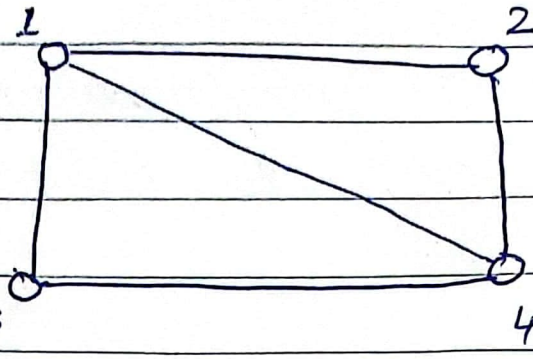
A CSP is a type of problem in AI where:
"You have to assign values to variables without breaking any rule (constraints)"

9×9=8 sudoku

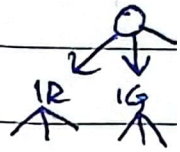


How CSP Algorithm works? NO Date . .

Backtracking (Intelligent)



Constraint Graph(



$V = \{1, 2, 3, 4\}$

$D = \{$ Red, Green, Blue $\}$

$C = \{1 \neq 2, 1 \neq 3, 1 \neq 4, 2 \neq 4, 3 \neq 4\}$

		1	2	3	4
	Initial Domain	RGB	RGB	RGB	RGB
Forward checking	1 = R	R	GB	GB	GB
Constraint propagation	2 = G	R	G	GB	B
	3 = G	R	G	G	B
Inference by thinking ahead					

It is a time efficient

Real world Examples of CSP

→ sudoku

→ Timetable scheduling

→ Map coloring

Local search in CSP

Instead of trying all combinations (like backtracking), local search starts with a random guess, then keeps changing it slightly to reduce rule violations (conflic

Min-conflicts Heuristic

A smart rule for local search:
Always pick the value for a variable that causes the fewest problems (conflicts)

How it works?

1. Choose a variable that's in conflic
2. Try all possible values
3. Pick one with minimum conflict.

Super fast practice for many CSPs.

NO Date . .

Constraint Weighting

When solving CSPs, some constraints are harder to satisfy.

So give them "weights" or importance score.

How it works:

i) Start with equal weights.

ii) Everytime a constraint is violated, increase its weight.

iii) The solver will pay more attention to solving this in future moves.