Learning in Online Search :

There exists several opportunities for learning provided by initial ignorance of online search agents.

✓ Agents learn a map of environment simply by recording each of their experiences.

✓ Most accurate estimates of the value of each state can be acquired by using local updating rules.

✓ These updates converge to exact values for each and every state provided that the agent explores the state space in the right way.

✓ Once the exact values are known, by moving to the highest-values successor, optimal decision can be taken.

2.2 CONSTRAINT SATISFACTION PROBLEMS (CSPS)

Constraint satisfaction problems are the problems whose goal test, states conform to a standard, structured and simple representation.

They are used to describe many real world problems. The standard representation of the goal tests, reveals the structure of the problem itself.

2.2.1 Introduction :

A constraint satisfaction problem is defined by a set of variables $X_1, \ldots X_n$ and a set of constraints $C_1, C_2, \ldots C_m$. It is also represented as CSP in short.

- Each variable X_i has non empty domain D_i of possible values.
- Each constraint C_i involves some subset of variables and allowable combination of values for the subset.
- A state is defined by assigning values to some or all of the variables $\{X_i = V_i \ldots\}$.
- Assignment can be a consistent one if it does not violate any constraints.
- An assignment can be a complete one if each variable is mentioned.
- A complete assignment satisfying all the constraints will be a solution to a CSP. A solution maximizing an objective function is required for some CSPs.

Constraint Graph :

A constraint graph is used to define the structure of a CSP.

Representation :

Nodes – Variables of the problem

Arcs – Constraints

Advantages or Benefits of Treating a problem as CSP :

✓ The successor function and goal test can be written in a generic way that applies to all CSPs.

✓ Effective, generic heuristics that require no additional domain-specific expertise can be developed.

✓ The structure of the constraint graph can be used to simplify the solution process.

A CSP can be given an incremental formulation as follows :

Initial State : empty assignment { }. All variables are unassigned.

Successor function : A value can be assigned to any unassigned variable.

Goal Test : Current assignment is complete

Path cost : A constant cost for every step.

A complete-state formulation in which every state is a complete assignment that might or might not satisfy the constraints can also be used.

Types of Variables in CSP :

Variables in CSP can be one of the following types namely

- discrete variables with finite domains
- Boolean CSPs
- discrete variables with infinite domains

Discrete variables with finite domains :

The simplest kind of CSPs involves variables that are discrete having finite domains.

Example : Map coloring problems.

The number of possible complete assignments is $0(d^n)$. Where d → maximum domain size of any variable.

Boolean CSPs :

Boolean CSPs are the finite-domain CSPs whose variables are either true or false.

Example : 3SAT, a NP complete problem.

Discrete Variables with Infinite Domains :

Discrete variables can also have infinite domain's such as set of integers or strings.

Example :

Scheduling problem of jobs on to a calendar. In case of infinite domains it is impossible to describe constraints by listing all allowed combination of values.

Hence constraint language must be used.

Example :

Start $Job_1 + 5 \leq$ start Job_3 is used to represent that a Job_1 taking 5 days must precede Job_3.

It is also no longer possible to solve these constraints by listing all possible assignments because they are infinite.

Special solution algorithms exist for linear constraints and no algorithm exists for solving non linear constraints on integer variables.

CSPs with continuous domains are common in real world. The best known category of continuous domain CSPs is linear programming problems where constraints must be linear inequalities. They can be solved in time polynomial in the number of variables.

Types of Constraints :

The following are the types of constraints that can appear in CSPs.

- Unary Constraints
- Binary Constraints
- Higher Order Constraints
- Absolute Constraints
- Preference Constraints

Unary Constraints :

- Simplest type of constraint.
- It restricts the value of a single variable.
- It can be eliminated simply by preprocessing the domain of the corresponding variable to remove any value that violates the constraint.

Example :

SA actively dislike the color green.

Binary Constraints :

- Relates two variables
- A binary CSP with only binary constraints can be represented as a constraint graph.

Example :

$$SA \neq NSW$$

Higher Order Constraints :

- These constraints involve three or more variables.

Example :

Cryptarithmetic puzzles in which each letter stands for a digit. The aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct with the constraint that no leading zeroes are allowed.

$$\begin{array}{rr} & \text{TWO} \\ + & \text{TWO} \\ \hline & \text{FOUR} \end{array}$$

The constraints can be written as

$$0 + 0 = R + 10.X_1$$

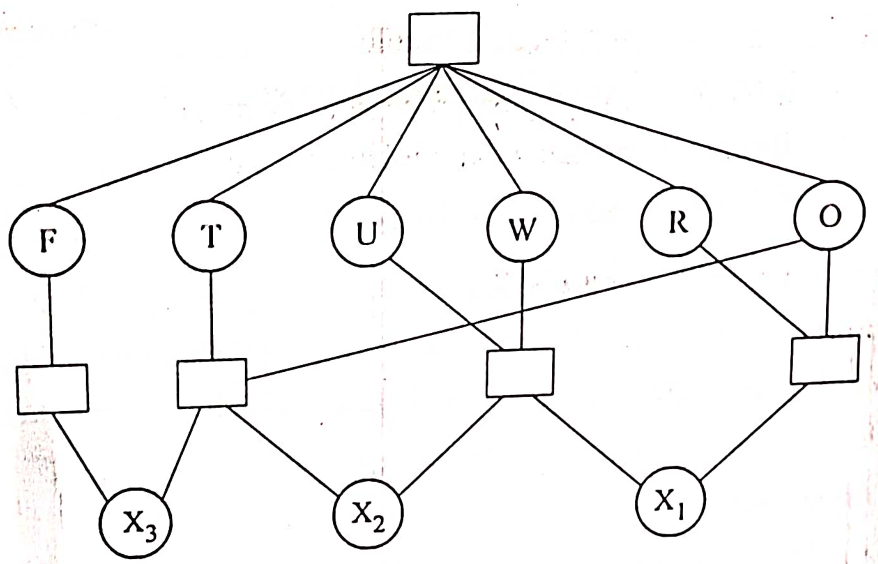
$$X_1 + w + w = U + 10.X_2$$

$$X_2 + T + T = 0 + 10.X_3$$

$$X_3 = F$$

X_1, X_2, X_3 are auxiliary variables.

It can be represented as constraint hypergraph in which each constraint is a square box connected to variables it constraints.



Absolute Constraints :

Absolute constraints are the constraints, which rules out a potential solution when they are violated.

Unary, Binary and higher-order constraints are absolute constraints.

Preference Constraints :

✓ Preference constraints are the constraints indicating which solutions are preferred.

Example : University Time Tabling Problem.

✓ It can be encoded as cost on individual variable assignments.

✓ CSPs with preferences can be solved using optimization search methods.

The general purpose methods can be found that addresses the following issues.

❖ Variable assignment and value ordering.

❖ Propagation of information through constraints.

❖ Intelligent backtracking.

Variable Assignment and Value ordering :

In order to decide which variable to be assigned next and in what order its values to be tried the following heuristics can be used.

- Minimum Remaining Values heuristic.
- Degree heuristic
- Least-constraining heuristic

Minimum Remaining Values Heuristic (MRV)

The SELECT-UNASSIGNED-VARIABLE function simply selects the next unassigned variable from the VARIABLES[CSP] list. This idea of choosing the variable with fewest legal values is called MRV heuristic.

Other Names :

"Most constrained variable" heuristic.

"fail first" heuristic

– *Reason :*

It picks a variable that is most likely to cause a failor soon.

Advantages :

✓ Results in most efficient search by avoiding pointless searches through other variables which always will fail.

Comparison of Various CSP Algorithms on Various Problems :

Problem	Backtracking	BT + MRV	Forward checking	FC+MRV	min-conflicts
USA	(>1000K)	(>1000K)	2K	60	64
n-queens	(>40,000K)	13,500K	(>40,000K)	817K	4K
Zebra	3,859K	1K	35K	0.5K	2K
Random 1	415K	3K	26K	2K	
Random 2	942K	27K	77K	15K	

The results suggest that FC + MRV is better on all these problems.

Disadvantage :

It doesn't help in choosing the first region to color in Australia because initially every region has three colors red, blue, green.

Degree heuristic :

This heuristic attempts to reduce the branching factor on future choices by selecting the variable that is involved in the largest number of constraints on other unassigned variables.

Advantages :

✓ This heuristic, solves the coloring problem without any false step.

✓ It acts as a tie-breaker

Least-constraining-value heuristic :

In order to determine the order in which to examine its values, Least-constraining value heuristic is used.