Constraint Satisfaction Problem

26/02/2024

Koustav Rudra

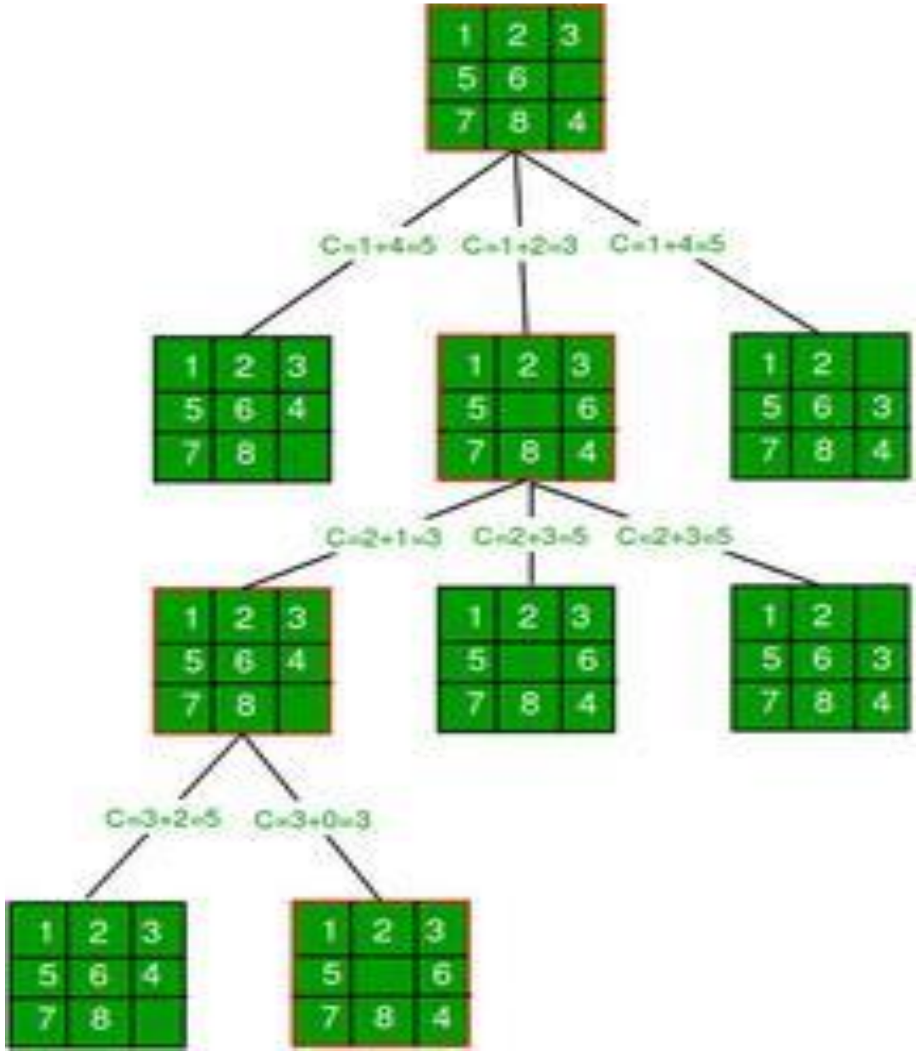
Objective

- Problem Formulation
- Problem representation
- Solvers

AI Problem Solvers: Evolution

```
def solve(State state):
        ………………………………….
        ………………………………….
        move(c1, c2)
        check(solution)
        ………………………………….
        ………………………………….
```

Brute-Force Approach



Problems:

- Very much problem specific
- Solution developed for one problem will not work for others

AI Problem Solvers: Evolution

```
def solve(State state):
        ……………………………………
        ……………………………………
        state.isGoal()
                return true
        succ = state.successor()
        ……………………………………
        ……………………………………
```

Search Algorithms

- **Overall Structure:** Problem Agnostic
- Still isGoal and successor are problem specific

- Can we have Truly Generic Problem Solvers?
- Yes, but for specific class of problems
  - Constraint Satisfaction Problems
- What are the implications?
  - Make isGoal and successor are problem agnostic
  - Design methods and heuristics: problem agnostic

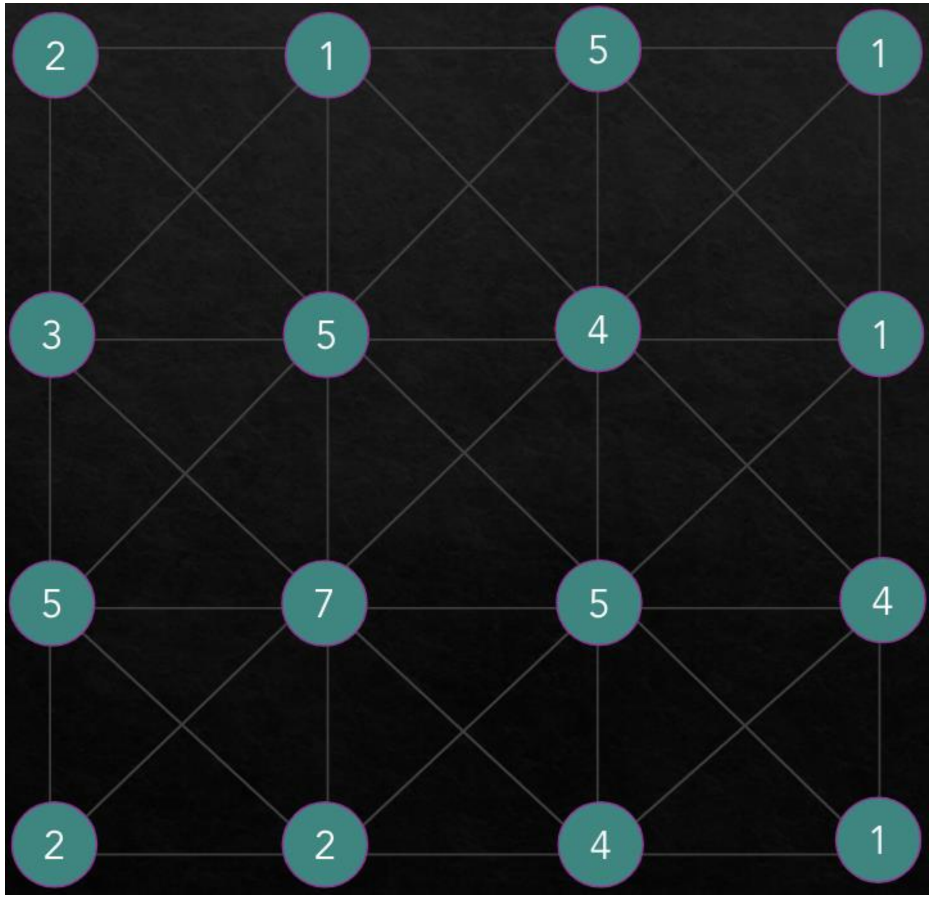
Revisiting Search Problems

- The world
  - Single agent, deterministic action, fully observable, discrete state

- Planning a sequence of actions
  - Important: Path to goal
  - Paths: varying costs and depths
  - Heuristics to reduce search space

- Identification of goal
  - Goal is important not path
  - All paths are at same depth
  - CSPs are identification problems

Example



Search Formulation:

1. Initial state: Nodes with no connection
2. Successor Function
    1. Add any one edge
    2. Next state: Resultant graph
3. Goal Test
    1. Whether each node has degree equal to the no. attached to the node

Path to Goal important?

Or

Configuration that satisfy certain criteria?

Constraint: Number of outgoing edges

Assignment: On/Off

Jointly all the assignments make sense or not

Combinatorial problem

Goal Identification Problem

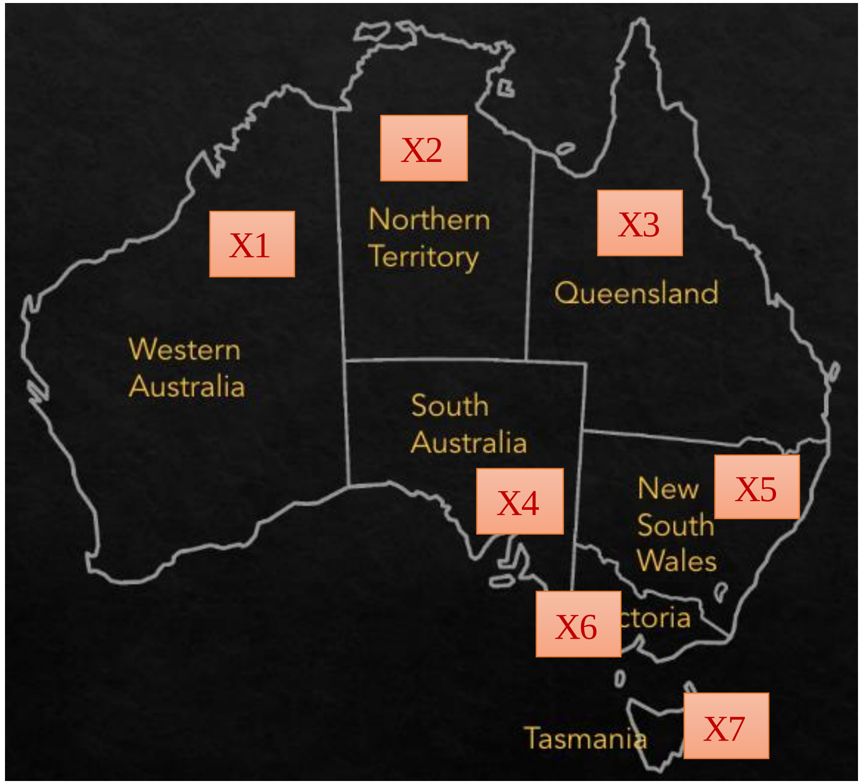
Can we define domain independent methods to solve the problem?

Constraint Satisfaction Problems

- Standard search problems:
  - State is problem independent ➔ Arbitrary data structures
  - Goal test: Function of state
    - Problem dependent
  - Successor: Function of state
    - Problem dependent

- Constraint Satisfaction Problems
  - Subset of search problems [Identification Problem]
  - State: $<Xi,Di>_N$
  - Goal Test: A set of constraints
    - $C1 \wedge C2 \ldots \wedge Cn$
    - Legal combination of values for subset of variables

Constraint Satisfaction Problems



Map Coloring Problem

- No two adjacent states have same color

CSPs: Formulation

- CSPs Problem: <X, D, C>

- State: X➔set of variables, Domain(Xi) = Di
  - X = $\{X_1, X_2, \ldots, X_n\}$
  - D = $\{D_1, D_2, \ldots, D_n\}$
- Goal Test:Set of constraints C
  - Ci = f(X') where $X' \subseteq$ X

- Constraint Definition
  - A pair <scope, rel>
  - Scope defines the variables
  - Relation describes interaction among variables in scope

- Example: X_1 and X_2 have domain {A, B}
  - Constraints: <(X_1,X_2), [(A,B), (B,A)]> [Explicit]
  - Constraints: <(X_1,X_2), $X_1 \neq X_2$ > [Implicit]

CSPs: Formulation

- Solution
  - Assignment: Assigning values to some or all variables
  - Consistent Assignment: Does not violate any constraint
  - Complete Assignment: Every variable is assigned a value
  - Solution: Consistent and Complete Assignment

- General purpose algorithms with more power than standard search algorithms

Constraint Satisfaction Problem

26/02/2024

Koustav Rudra

Example: Sudoku

	1	2	3	4	5	6	7	8	9
A		6		1		4		5	
B			8	3		5	6		
C	2								1
D	8			4		7			6
E			6				3		
F	7			9		1			4
G	5								2
H			7	2		6	9		
I		4		5		8		7	

Variables: Each open square

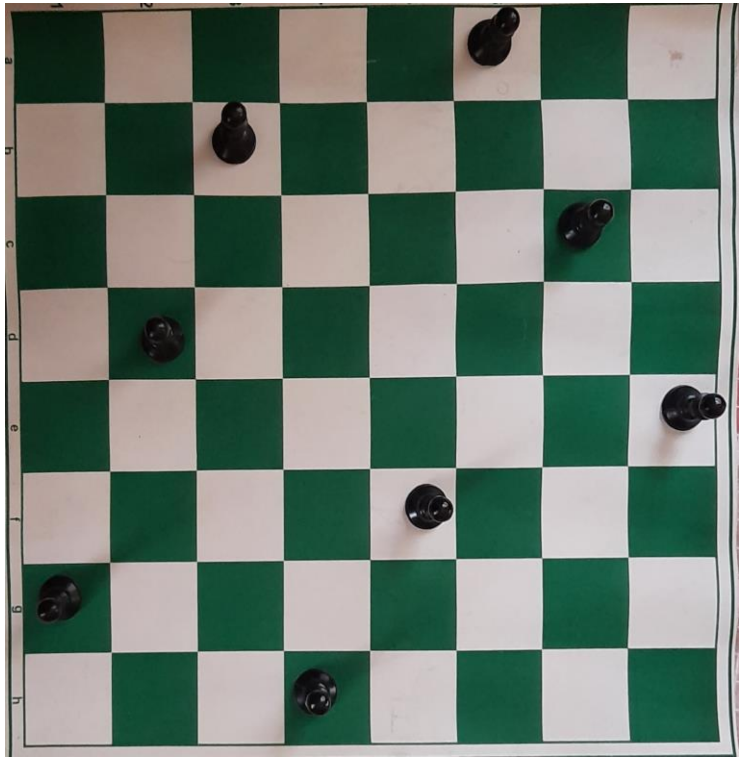
Domain: {1,2,3,4,5,6,7,8,9}

- Constraint
  - 9 ways all different for columns
  - 9 ways all different for rows
  - 9 ways all different for regions
- Constraint
  - <A11 ≠ A12, A11 ≠ A13,…,A11 ≠ A19>
  - <A12 ≠ A13, A12 ≠ A14,…,A12 ≠ A19>

Example: Map Coloring

- Variables: {WA, NT, SA, Q, NSW, V, T}
- Domain: {blue, red, green}
- Constraint: Adjacent regions have different colour
  - {WA≠NT} or
  - (WA, NT) ∈{(red, green), (red, blue), …}

Example: N-Queens



- Variables: $\{X_{ij}\}$ij∈{1,…,8}

- Domain: {0,1}

- Constraint: $(X_{11}, X_{12}) \in \{(0,0),(1,0),(0,1)\}$
  - $\forall_{i,j,k}(X_{ij}, X_{jk}) \in \{(0,0),(1,0),(0,1)\}$
  - $\forall_{i,j,k}(X_{ij}, X_{kj}) \in \{(0,0),(1,0),(0,1)\}$
  - $\forall_{i,j,k}(X_{ij}, X_{i+kj+k}) \in \{(0,0),(1,0),(0,1)\}$
  - $\forall_{i,j,k}(X_{ij}, X_{i-kj-k}) \in \{(0,0),(1,0),(0,1)\}$
- $\sum_{ij} X_{ij} = N$

Thank You