Artificial Intelligence

Fiona Yan Liu
Department of Computing
The Hong Kong Polytechnic University

Learning Outcomes of Heuristics

- An admissible heuristic never overestimates the cost to reach the goal
  - A heuristic $h(n)$ is admissible if for every node n, $h(n) \leq h^*(n)$, where $h^*(n)$ is the true cost to reach the goal state from n
  - The tree-search version of A* is optimal if $h(n)$ is admissible
- Domination of heuristic functions
  - If $h_2(n) \geq h_1(n)$ for all n (both admissible)
  - then h_2 dominates h_1, which indicates that h_2 is better for search
- Local search
  - Hill-climbing search
  - Simulated annealing search
- Reference reading
  - Chapter 4

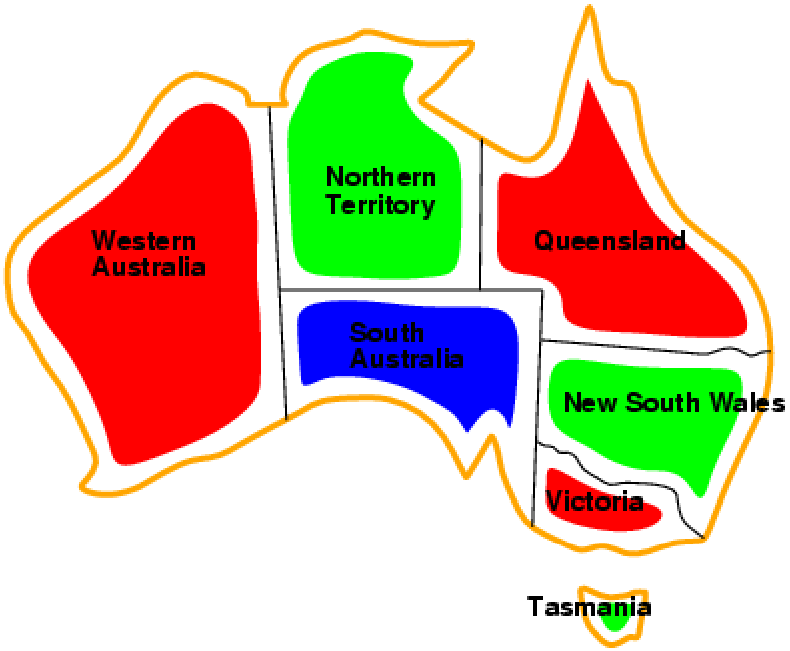
Constraint Satisfaction Problems

- Standard search problem
  - Each state is atomic with no internal structure
  - These states can be evaluated and test to see whether they are goal states
- Constraint satisfaction problem (CSP)
  - State is defined by variables X_i with values from domain D_i
  - Goal test is a set of constraints specifying allowable combinations of values for subsets of variables
- CSP consists of three components
  - A set of variables
  - A set of domains
  - A set of constrains
- CSP enable solutions of complex problems

Example: Map-Coloring

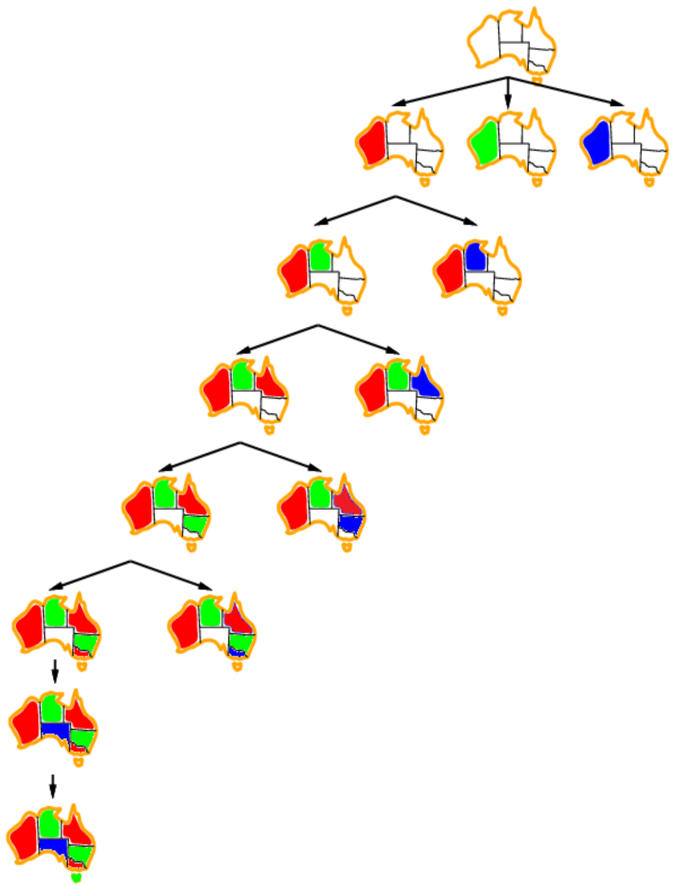
- Variables {WA, NT, Q, NSW, V, SA, T}
- Domains Di = {red,green,blue}
- Constraints: adjacent regions must have different colors
- e.g., WA ≠ NT, or (WA,NT) in {(red,green),(red,blue),(green,red), (green,blue),(blue,red),(blue,green)}

Example: Map-Coloring



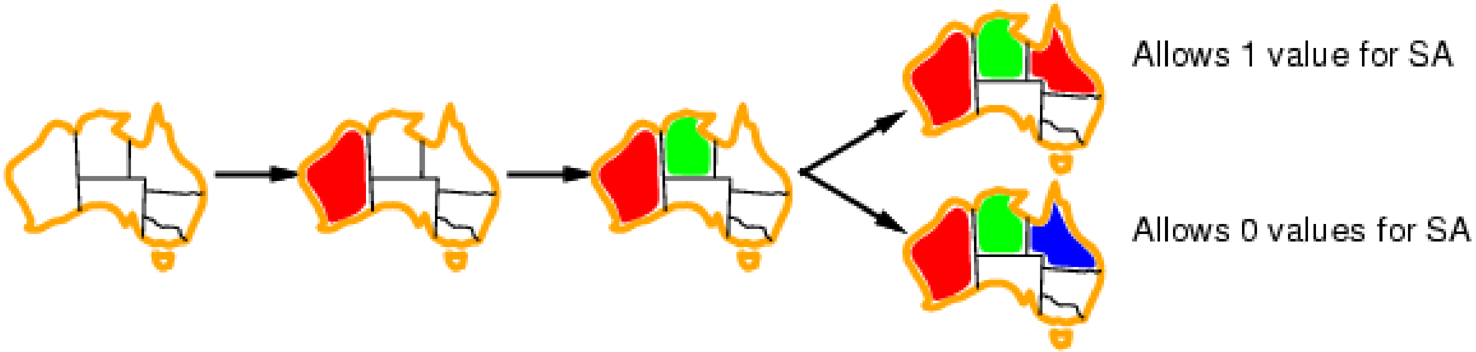
- Solutions are complete
- WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

Search for Map-Coloring



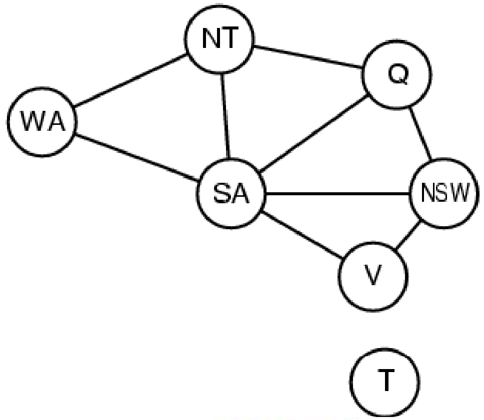
Least Constraining Value

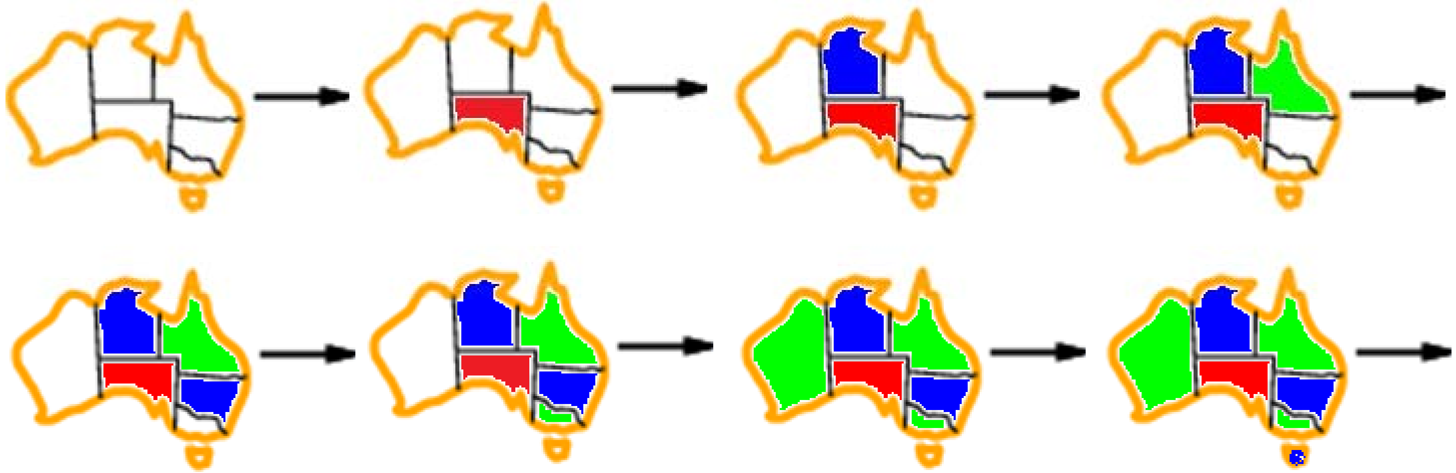
- Least constraining value
  - Given a variable, choose the least constraining value



Most Constrained Variable

- Constraint graph
  - nodes are variables, arcs are constraints
- Most constrained variable
  - Choose the variable with the fewest legal values





Applications

- Proposed by Michael R. Garey in 1976
  - For testing printed circuit boards
- Many scheduling problems can be formulated as a map coloring problem
  - Exam and class scheduling
  - Class scheduling
  - Job scheduling
  - Air traffic flow management
  - Bandwidth allocation in wireless communication
  - Register allocation in computer compiler

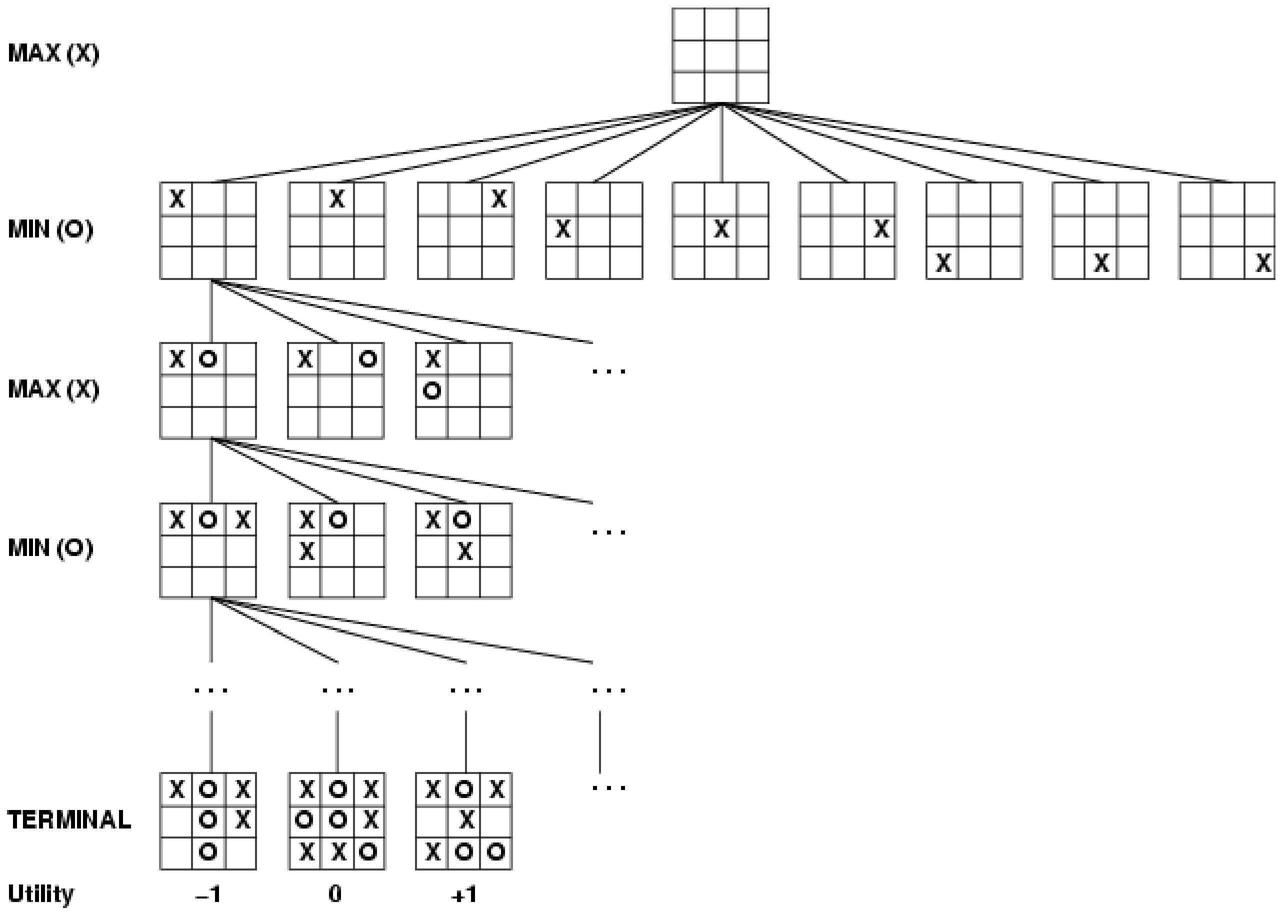
Games

- Multiagent environment
  - In which each agent needs to consider the actions of other agents
  - And how they affect its own welfare
- Games
  - Views any multiagent environment as a game, provided that the impact of each agent on the other is significant
- Games in AI
  - The state of a game is easy to represent
  - Agents are usually restricted to a small number of actions
  - Outcomes are defined by precise rules

Formulate Game to Search Problem

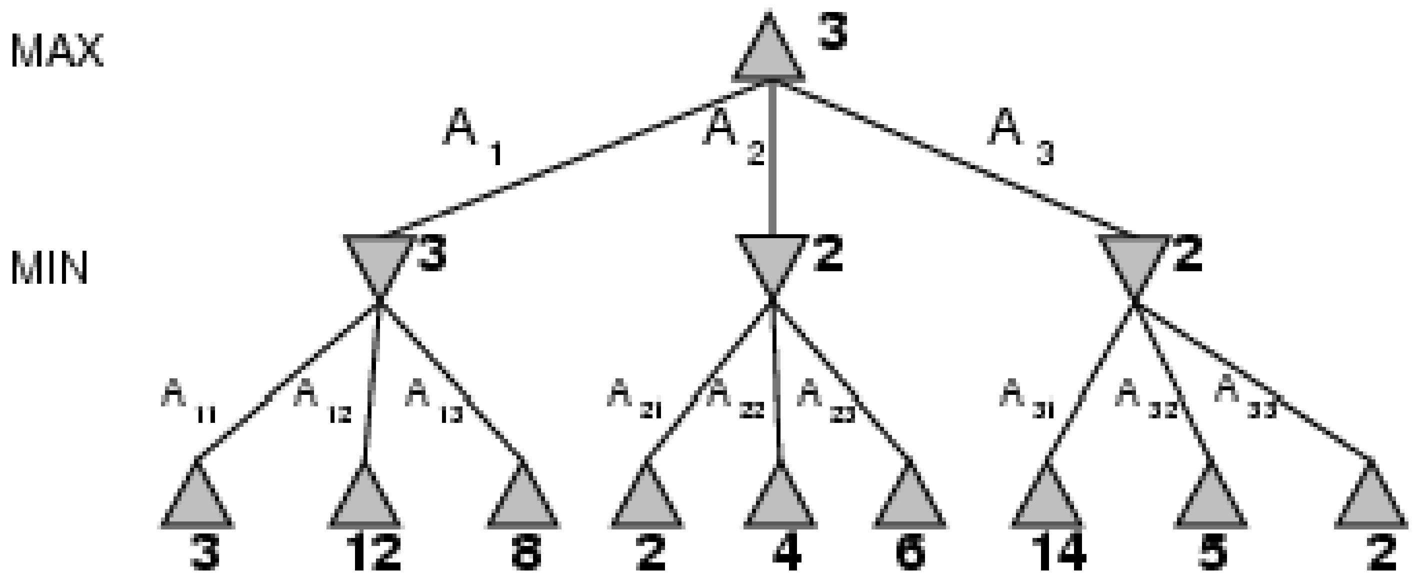
- S_0
  - The initial state, which specifies how the game is set up at the start
- Player(s)
  - Defines which players has the move in a state
- Actions(s)
  - Defines a set of legal move in a state
- Result(s, a)
  - The transition model, which defines the result of a move
- Terminal-Test(s)
  - A terminal test, which is true when the game is over
- Utility(s, p)
  - A utility function, also called an objective function or payoff function
  - Defines the final numeric value for a game that ends in terminal state s for a player p

Example of a Partial Game Tree



Optimal Decision in Games

- Perfect play for deterministic games
  - Idea: choose move to position with highest minimax value
  - best achievable payoff against best play



α-β Pruning