Soft Computing Introduction

N Geetha
A M & C S
P S G College of Technology

AI : Definition

- *AI may be defined as the branch of computer science that is concerned with the automation of intelligence behavior"* (G.F. Luger and W.A. Stubblefield)
- *"AI is the study of computations that make possible to behave intelligently"* (Patrick H. Winston)
- *"AI is the design and study of computer programs that behave intelligently"* (Thomas Dean, et al)

Task Domains of AI

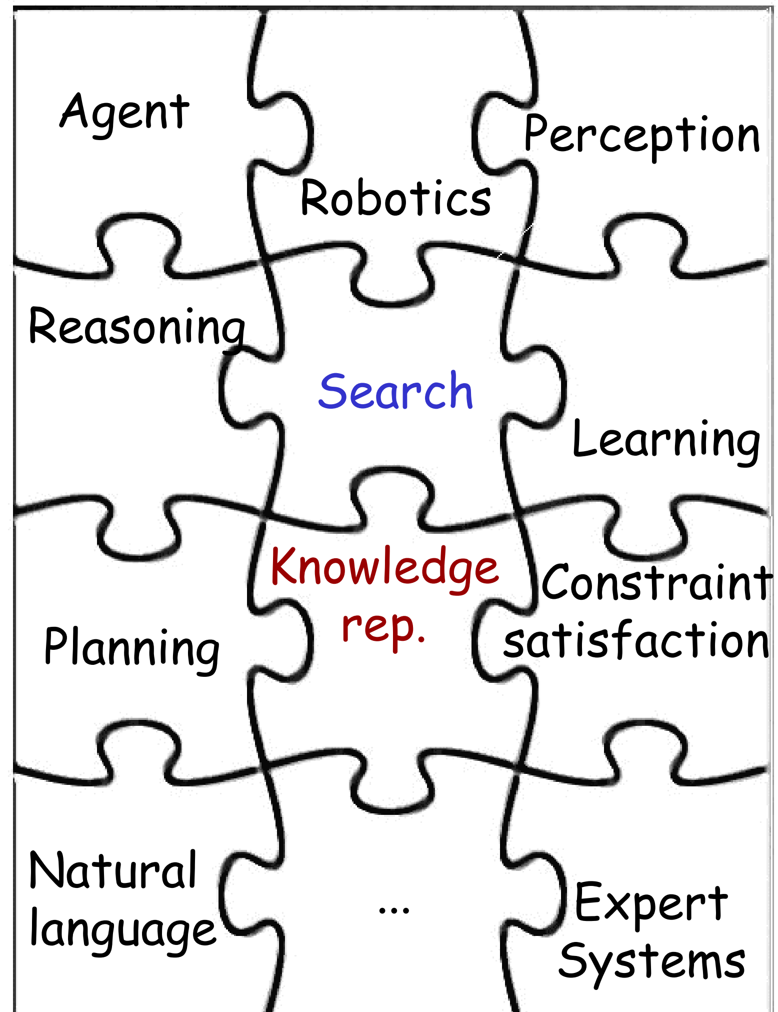
- **Mundane Tasks :**
  - Perception
  - Natural language
  - Commonsense reasoning
  - Robot Control
- **Formal Tasks :**
  - Games
  - Mathematics
- **Expert Tasks :**
  - Engineering
  - Scientific Analysis
  - Medical Diagnosis
  - Financial Analysis

Abridged History of AI

• 1943	McCulloch & Pitts: Boolean circuit model of brain
• 1950	Turing's "Computing Machinery and Intelligence"
• 1956	Dartmouth meeting: "Artificial Intelligence" adopted
• 1950s	Early AI programs, including Samuel's checkers
• 1965	Robinson's complete algorithm for logical reasoning
• 1966—73	AI discovers computational complexity Neural network research almost disappears
• 1969—79	Early development of knowledge-based systems
• 1980--	AI becomes an industry
• 1986--	Neural networks return to popularity
• 1987--	AI becomes a science
• 1995--	The emergence of intelligent agents

Main Areas of AI

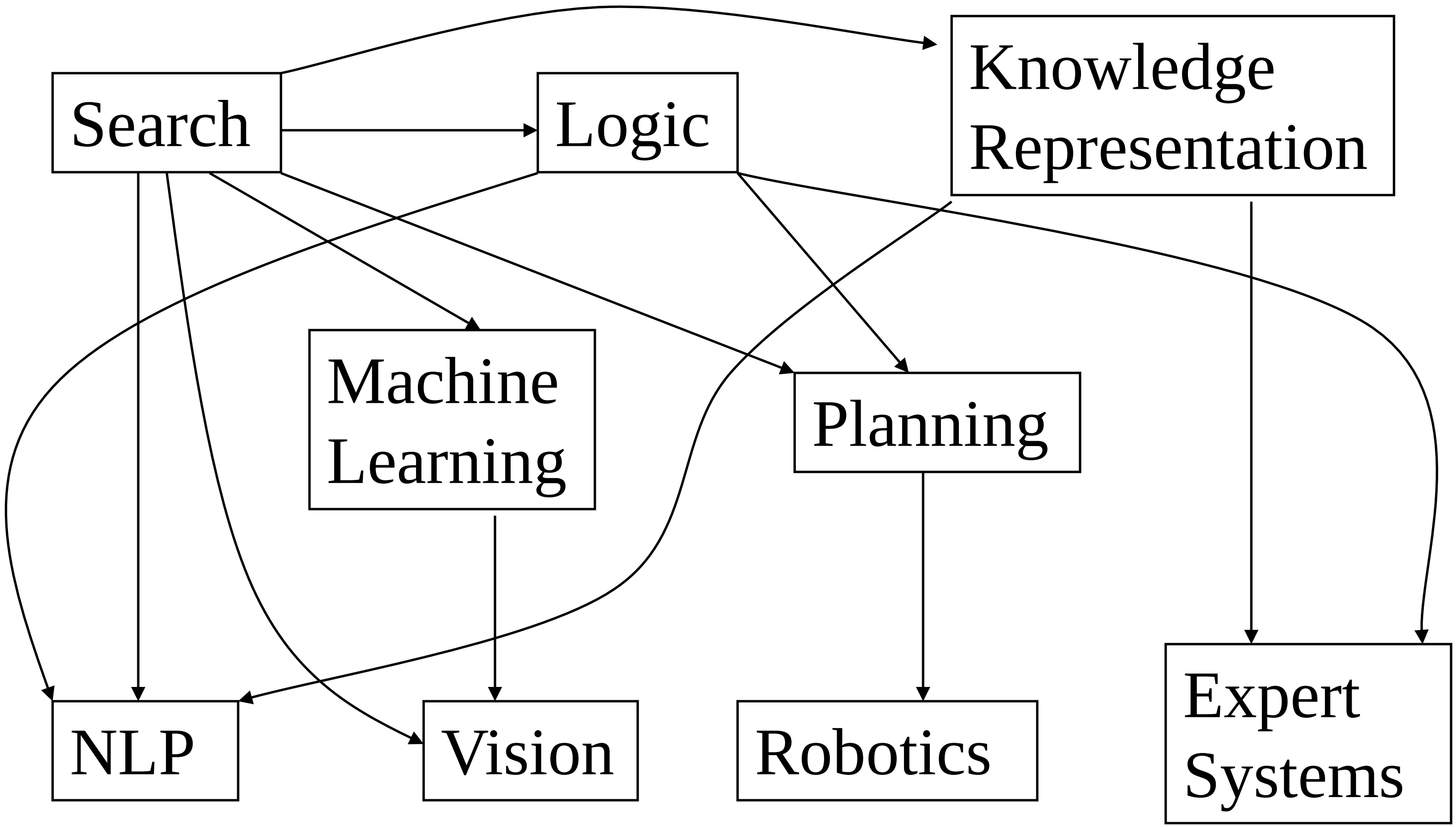
- Knowledge representation (including formal logic)
- Search, especially heuristic search (puzzles, games)
- Planning
- Reasoning under uncertainty, including probabilistic reasoning
- Learning
- Agent architectures
- Robotics and perception
- Natural language processing



Basic Techniques

- **Knowledge Representation**
  - ✓ Encoding and manipulating domain specific knowledge
  - ✓ Methods: Propositional Logic, Predicate Logic, Semantic Nets, Frames, Scripts
- **Searching**
  - ✓ Finding solutions among alternatives
  - ✓ Brute Force Search (uninformed search)
    - Generate and test every possibility
    - Often inefficient and sometimes infeasible
  - ✓ Heuristic Search (Informed Search)
    - Guiding the search with domain knowledge
    - More likely to lead to a good solution

Areas of AI and Some Dependencies



Search

- Search is a problem solving technique that systematically explores the space of problem states
- This space of alternative solutions is then searched to find an answer
- This is the essential basis of human problem solving
- Problem formulation is the process of deciding what actions and states to consider
- Go thru the set of states; Adopt a goal and aim to satisfy it

Search and AI

- Search methods are ubiquitous in AI systems. They are often the backbones of both core and peripheral modules
- An autonomous robot uses search methods:
  - To decide which actions to take and which sensing operations to perform
  - To quickly anticipate collision
  - To plan trajectories
  - To interpret large numerical datasets provided by sensors into compact symbolic representations
  - To diagnose why something did not happen as expected, etc.
- Many searches may occur concurrently and sequentially

Search and games

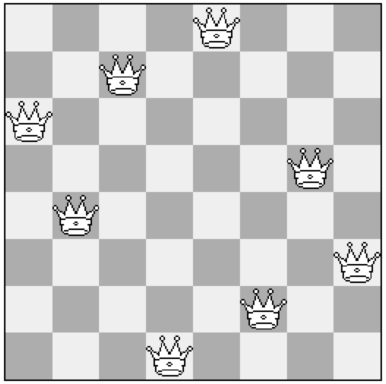
- Across history, puzzles and games requiring the exploration of alternatives have been considered a challenge for human intelligence
  - Chess originated in Persia and India about 4000 years ago
  - Checkers appeared in 3600-year-old Egyptian paintings
  - ‘Go’ originated in China over 3000 years ago
- So, it’s not surprising that AI uses games to design and test algorithms

Search problems

15 puzzle

Towers of Hanoi

Mazes



8 queens

Real life search problems

Scheduling

Route planning

Telescope scheduling

Military robot navigation

More applications

- Route finding
  - Airline travel, trains, roads etc.
  - Computer networks
- Package/mail distribution
- Pipe routing, VLSI routing
  - E.g. how to route cables and wires in a building, a car etc, or how to connect the pins of electronic chips
- Pharmaceutical drug design
  - How to find the most suitable substances that allow the active component of the drug to link to the target zone
- Video games

Search Methods

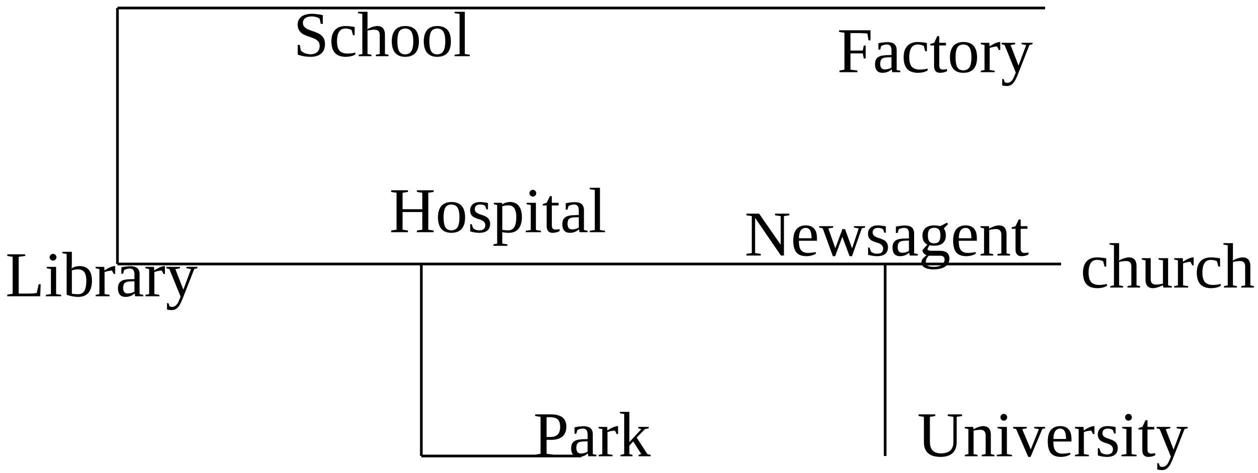
- Why search?
- Uninformed search methods
  - Breadth-first, depth-first, bidirectional, iterative deepening search
- Informed search methods
  - Greedy Hill Climbing, Best First, A Algorithm, A* search
- Conclusions

Well defined Problem

- **Initial** state is the starting position
- **Operator** is the action applied to move from one state to another. Given a state x, Succ(x) returns a set of (action, succ) ordered pairs where each action is one of the legal actions in stat x, and each successor is a state that can be reached from x by applying the action.
- **State Space** : Set of states reachable from the initial state by any sequence of actions
- **Goal** is the final state to reach
- A **path** in the state space is a sequence of actions leading from one state to another
- **Goal test** is applied to determine if it is a goal state or not
- **Path Cost** function is a function that assigns a cost to a path
- **Solution** to the problem is a path from the initial state to a goal state.

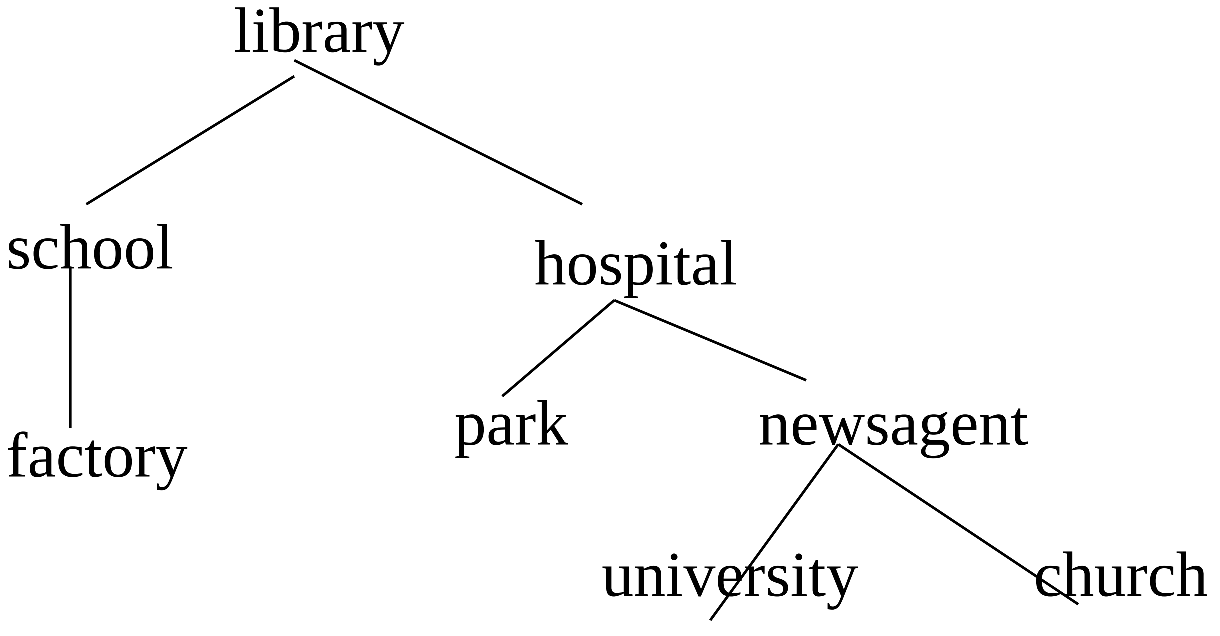
Simple Example

- Easiest to first look at simple examples based on searching for route on a map.
- How do we systematically and exhaustively search possible routes, in order to find, say, route from library to university?



Search Space

- The set of all possible states reachable from the initial state defines the *search space*.
- We can represent the search space as a tree.
- We refer to nodes connected to and "under" a node in the tree as "successor nodes".

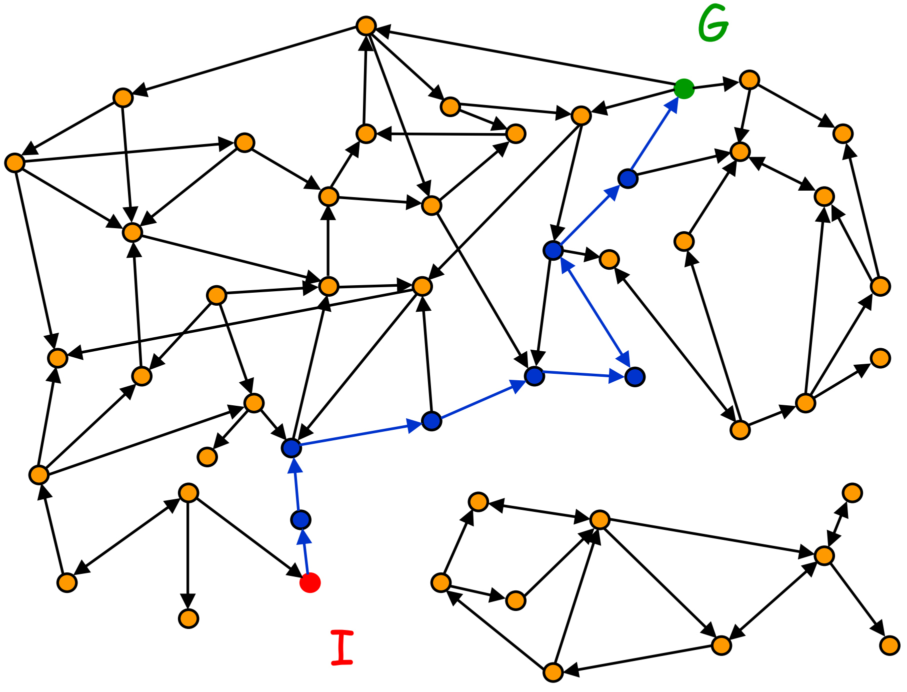


Formalization of a search problem

- **Q** is a finite set of states
- **S** $\subseteq$ Q is a non-empty set of start states
- **G** $\subseteq$ Q is a non-empty set of goal states
- **succs** : Q $\rightarrow \wp$(Q) is a function which takes a state as input and returns a set of states as output
  - succs(s) is the set of states that can be reached from s in one single step
- **cost** : $Q^2 \rightarrow R^+$ is a function which takes two states as input, and returns a positive number
  - cost(s,s') is the cost of getting from s to s'
  - cost(s,s') is only defined when s' $\in$ succs(s)

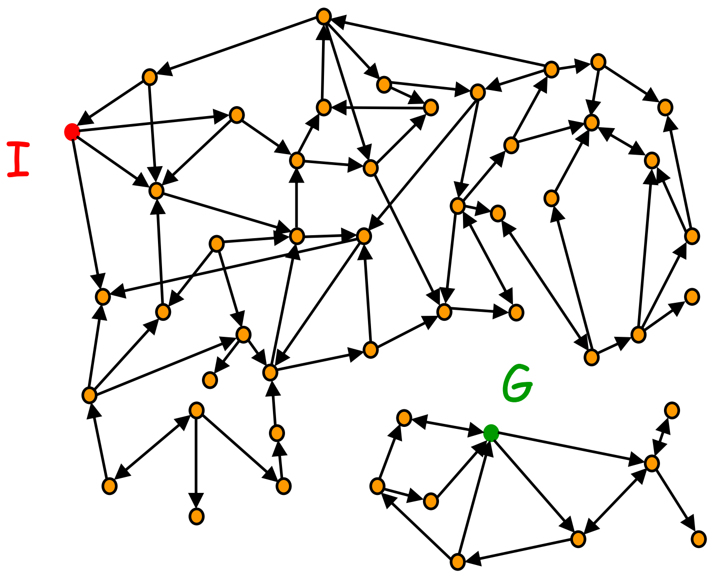
Solution to the search problem

- A solution is a path connecting the initial node to a goal node (any one)



Solution to the search problem

- A solution is a path connecting the initial node to a goal node (any one)
- The cost of a path is the sum of the arc costs along this path
- An optimal solution is a solution path of minimum cost
- There might be no solution!

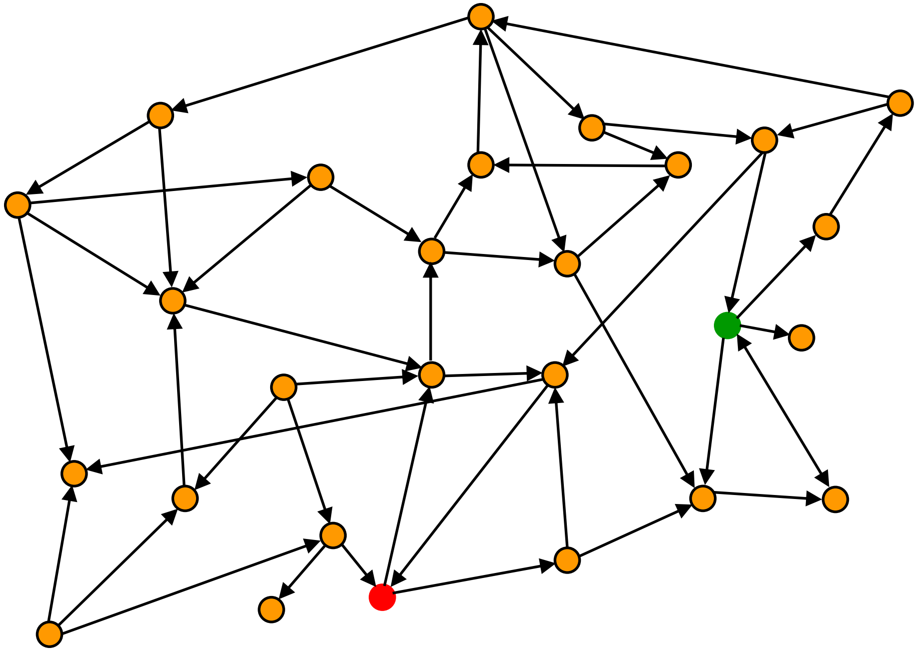


How big is the state space of the puzzle?

- 8-puzzle → 9! = 362,880 states
- 15-puzzle → 16! ≈ 2.09 x 10^{13} states
- 24-puzzle → 25! ≈ 10^{25} states

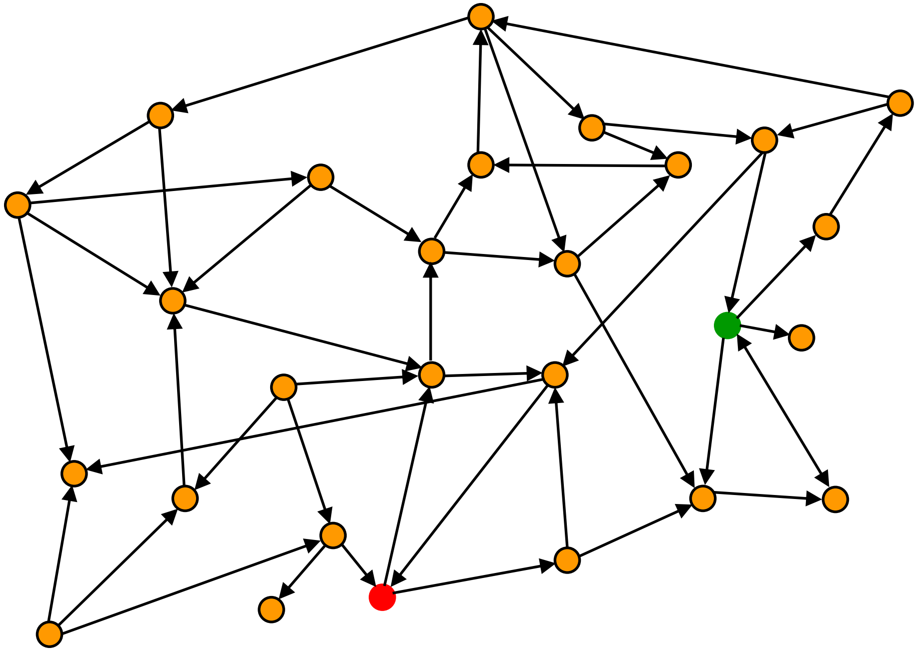
- Only half of these states are reachable from any given state
  - But that may not be known in advance

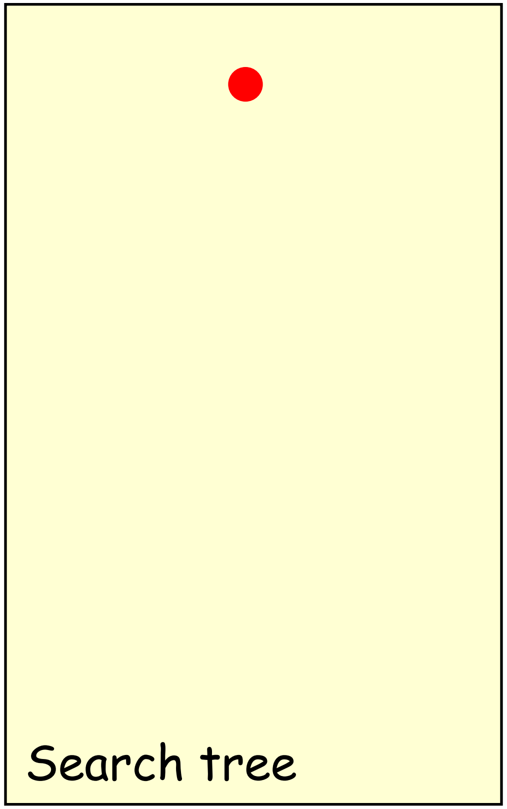
Searching the state space



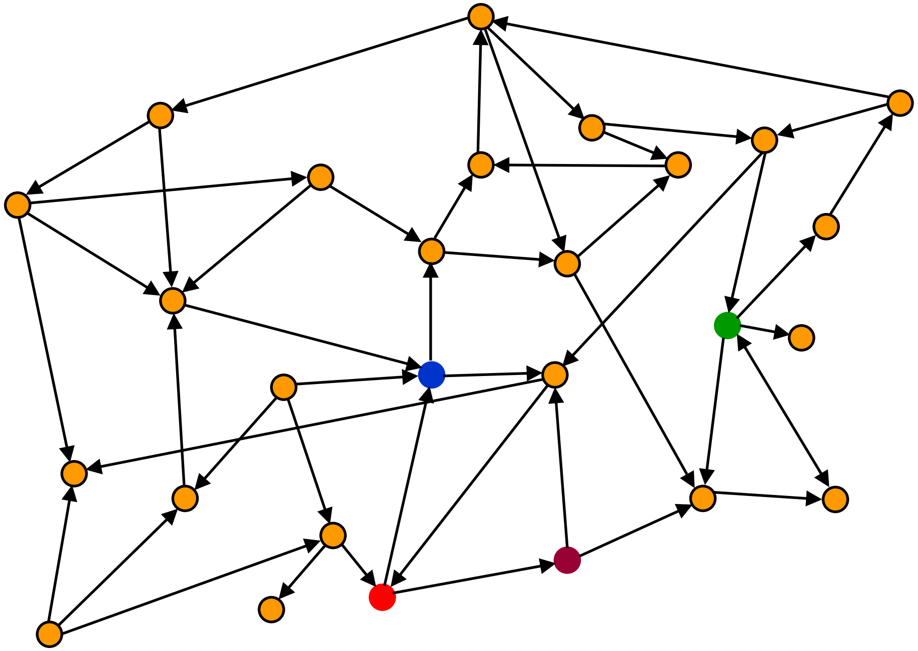
- Often it is not feasible (or too expensive) to build a complete representation of the state graph
- A problem solver must construct a solution by exploring a small portion of the graph

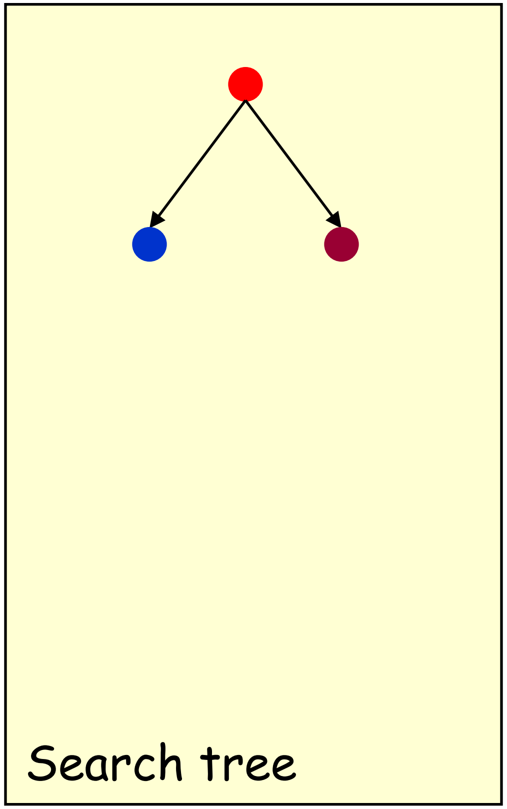
Searching the state space



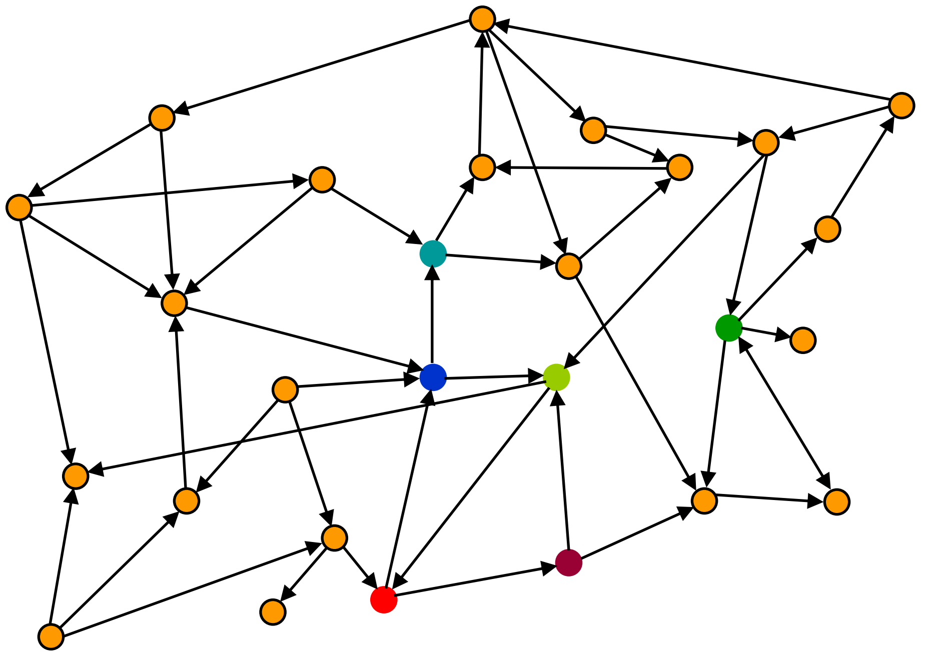


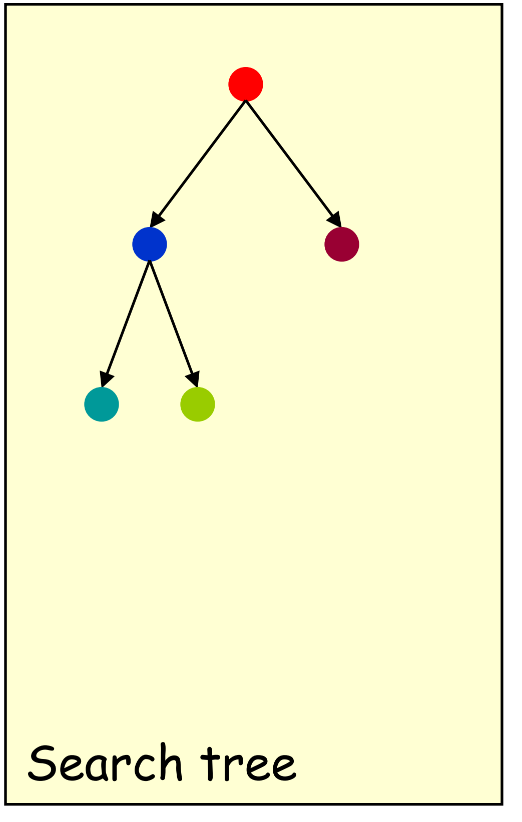
Searching the state space



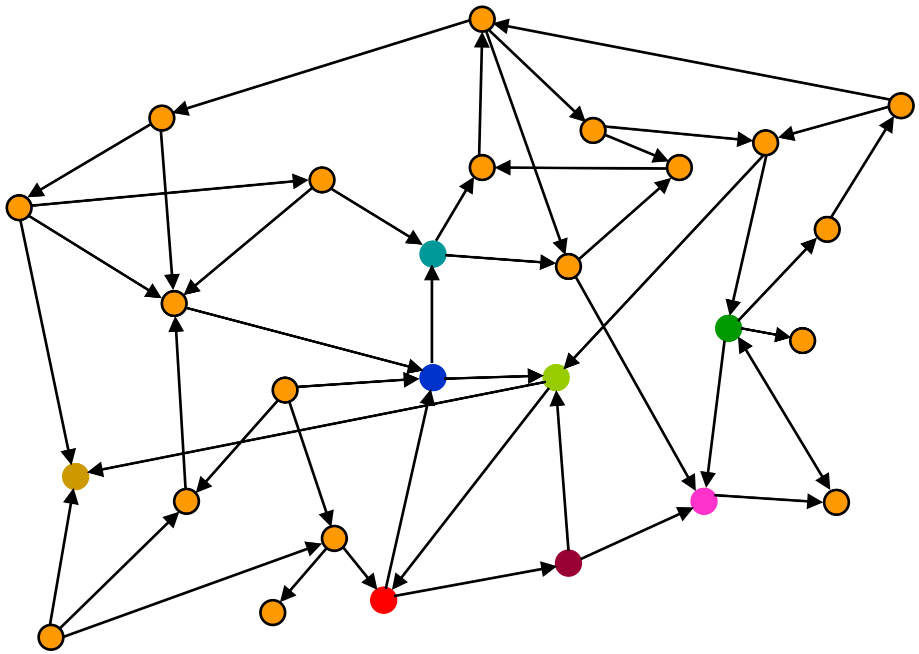


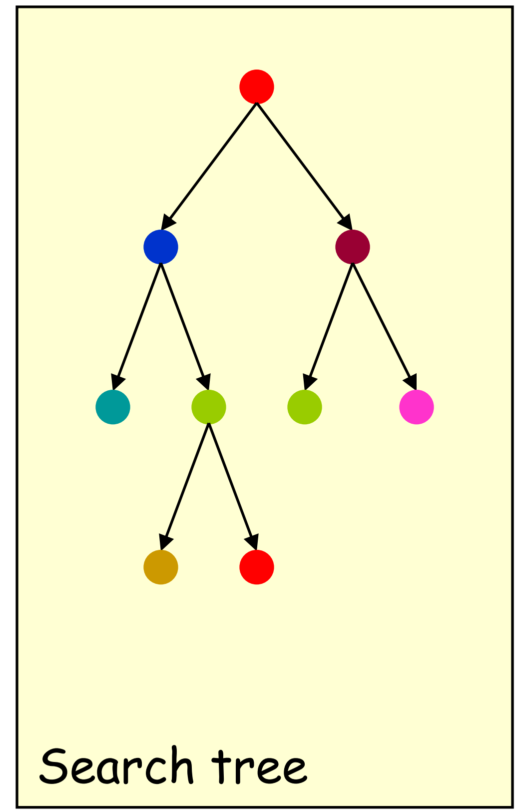
Searching the state space



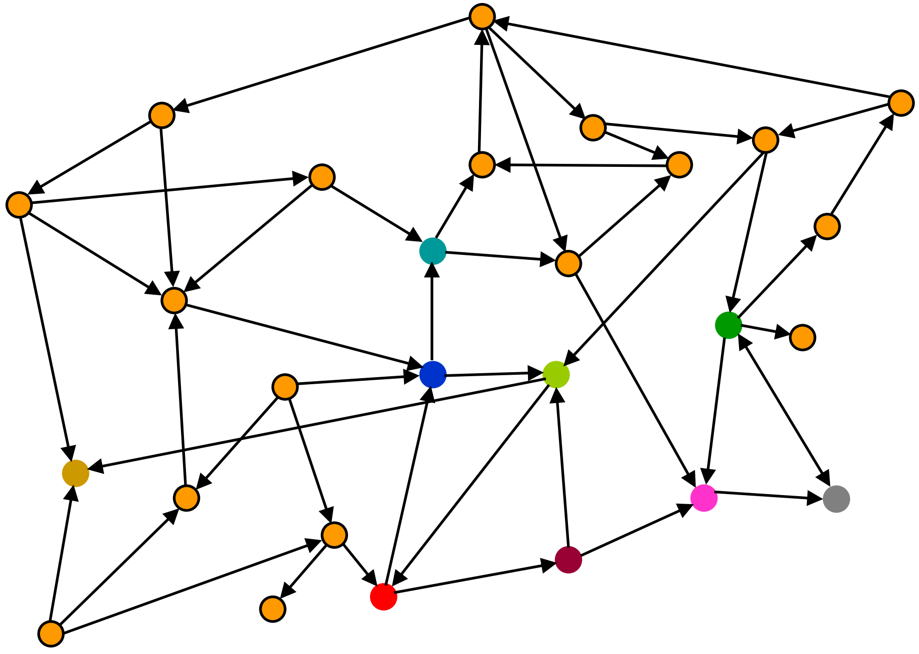


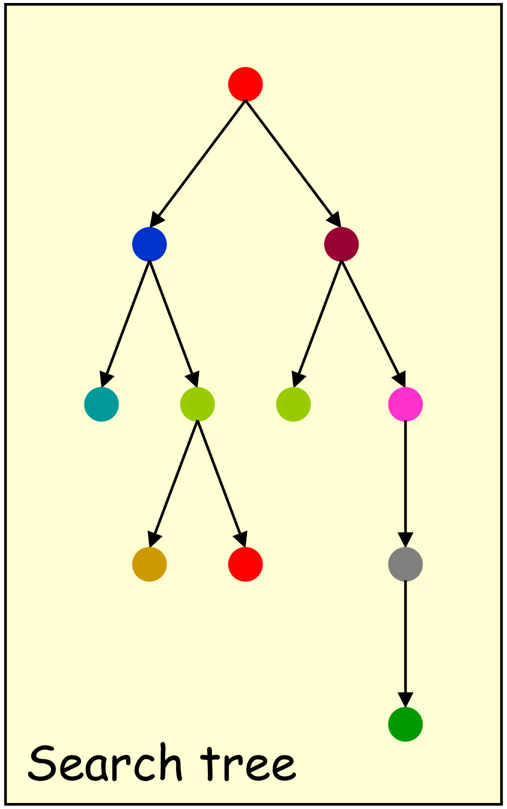
Searching the state space



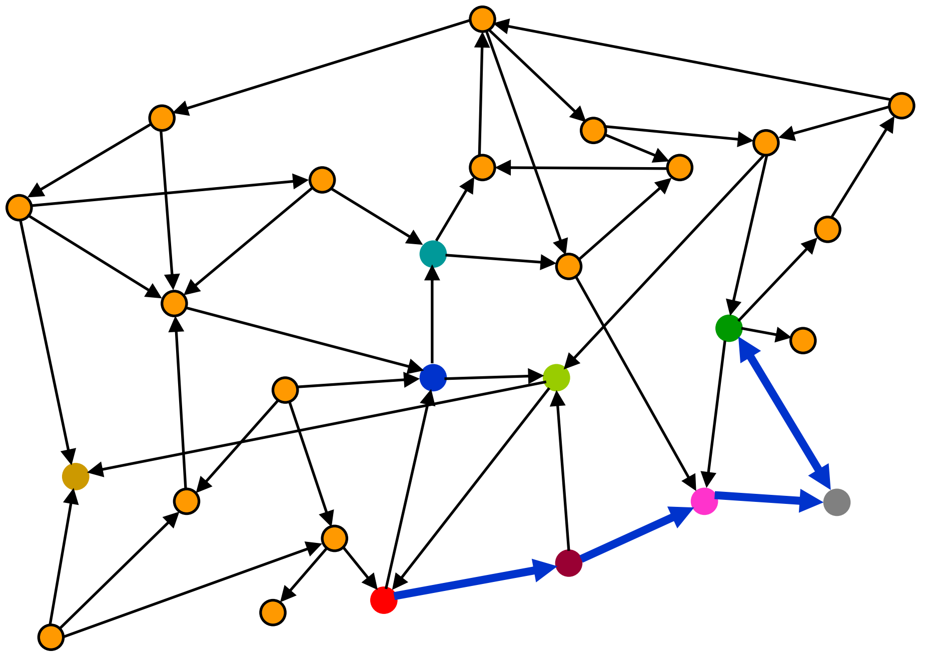


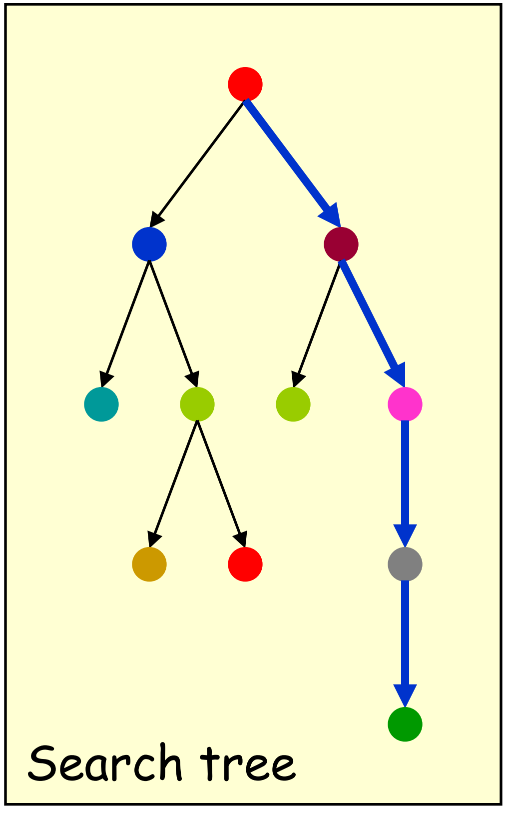
Searching the state space





Searching the state space





Simple Search Techniques

- How do we search this tree to find a possible route from library to University?
- May use simple systematic search techniques, which try every possibility in systematic way.
- Breadth first search - Try shortest paths first.
- Depth first search - Follow a path as far as it goes, and when reach dead end, backup and try last encountered alternative.

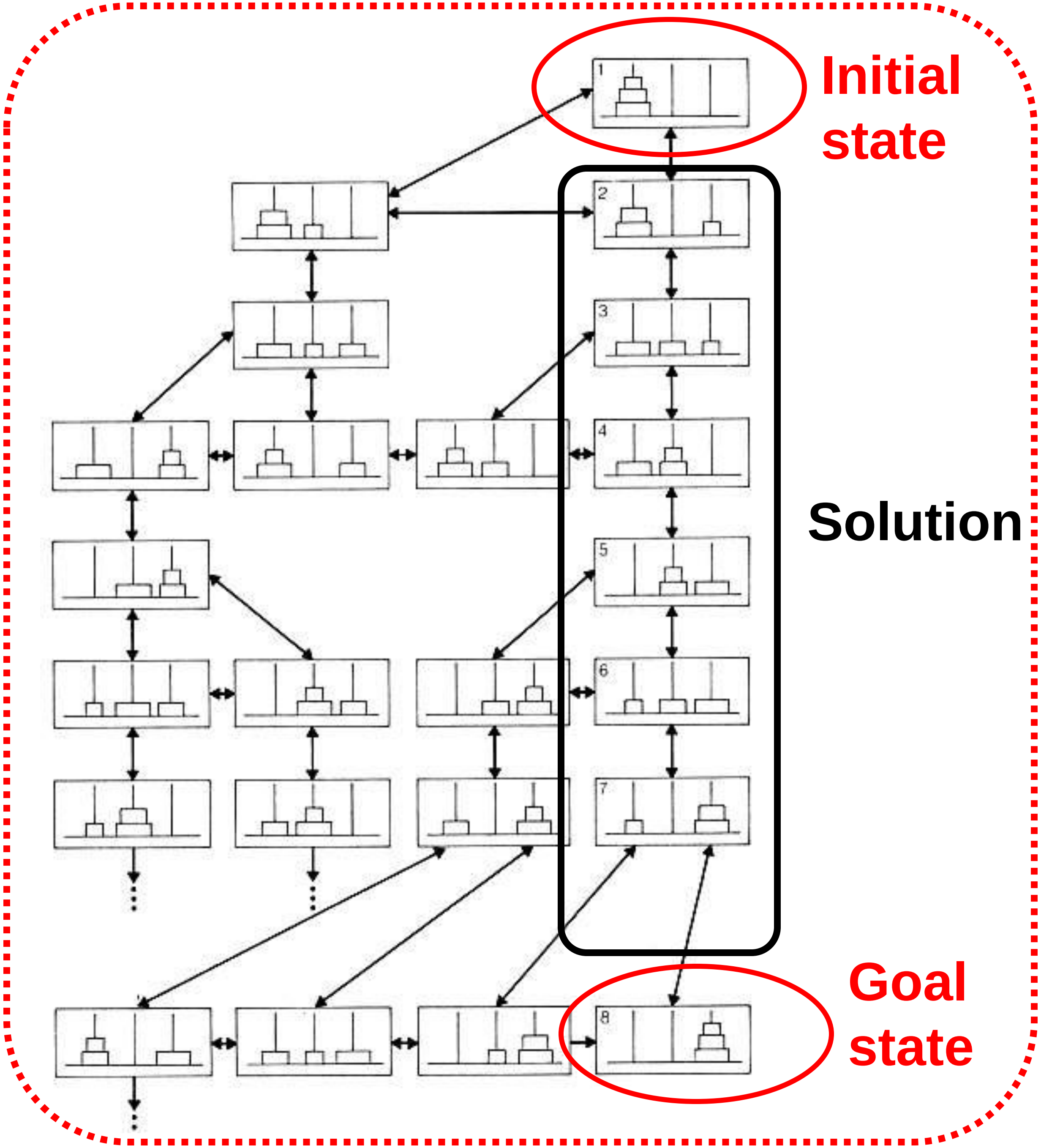
Solution to the Problem

- **Search space (Problem space)**: the set of all states that can potentially be reached by applying available operators

- **Solution**: a sequence of operators that transform the initial state into a goal state

- **A problem solving method**: a procedure for finding a solution

Problem Solving is *a search problem*



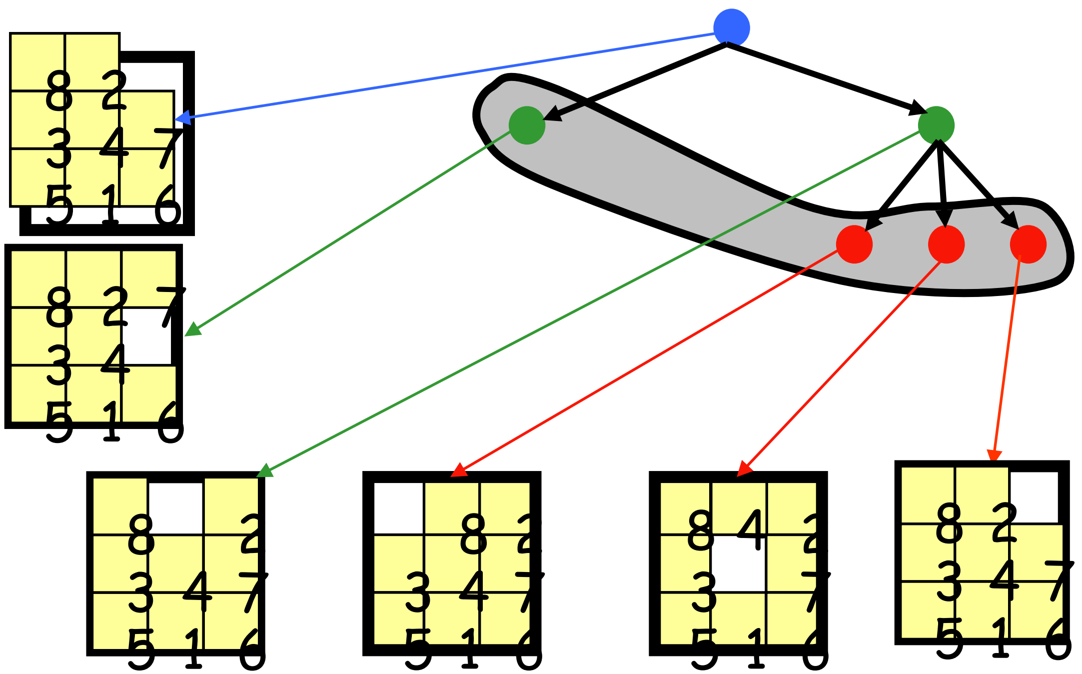
Search Space (Problem space)

Performance measures

- **Completeness**
  - A search algorithm is complete if it finds a solution whenever one exists
  - What about the case when no solution exists?
- **Optimality**
  - A search algorithm is optimal if it returns a minimum-cost path whenever a solution exists
- **Complexity**
  - It measures the time and amount of memory required by the algorithm

Fringe of a search tree

- The *fringe* is the set of all search nodes that haven't been expanded yet



Conclusion

- Search methods are useful when:
  - The search space is small, and
    - No other technique is available, or
    - Developing a more efficient technique is not worth the effort
  - The search space is large, and
    - No other available technique is available, and
    - There exist “good” heuristics

References

- Elaine Rich, Kevin Knight, “Artificial Intelligence”,Tata Mcgraw Hill, 2002.
- Amit Konar, ”Artificial Intelligence and Soft Computing”, CRC Press, 2000.
- Peter Norvig, Stuart Russel, “Artificial Intelligence: A modern Approach”, Prentice Hall of India, 2006.
- Florin Leon, http://eureka.cs.tuiasi.ro/~fleon