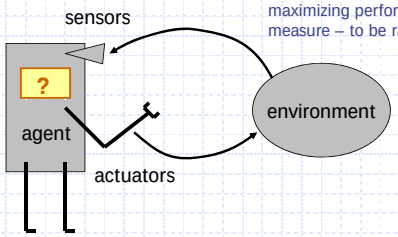
Search Problems

Russell and Norvig:
Chap. 3, Sect. 3.1 – 3.2

Slides by Jean-Claude Latombe, from an introductory AI course given at Stanford University Winter 2004 (used with permission).

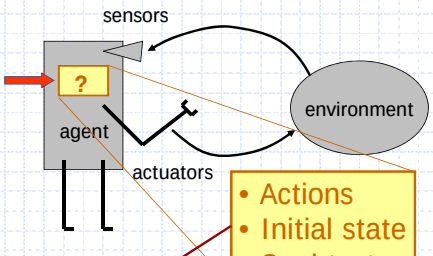
Goal-Based Agent

Goal: One way towards maximizing performance measure – to be rational



Can it find a sequence of actions achieving its goals, when no single action will do?

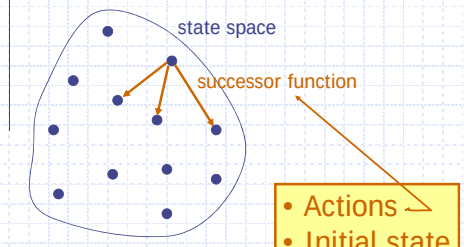
Problem-Solving Agent



- Actions
- Initial state
- Goal test

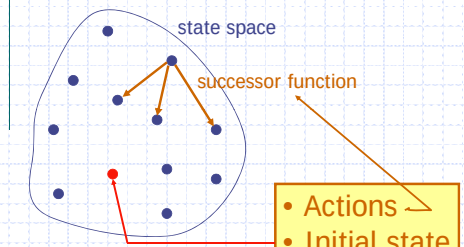
Graph searching

State Space and Successor Function



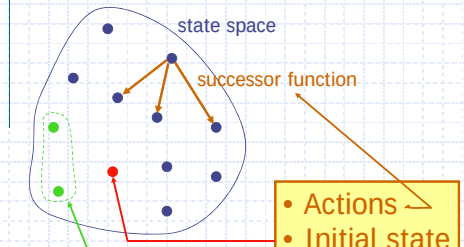
- Actions
- Initial state
- Goal test

Initial State



- Actions
- Initial state
- Goal test

Goal Test



- Actions
- Initial state
- Goal test

Example: 8-puzzle

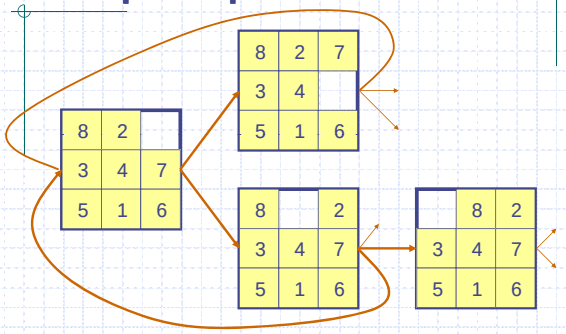
8	2	
3	4	7
5	1	6

Initial state

1	2	3
4	5	6
7	8	

Goal state

Example: 8-puzzle



Example: 8-puzzle

Size of the state space = 9!/2 = 181,440

15-puzzle → .65 x 10^{12}

24-puzzle → .5 x 10^{25}

0.18 sec

6 days

12 billion years

10 millions states/sec

Search Problem

- State space
- Initial state
- Successor function
- Goal test
- Path cost

Search Problem

- State space
  - each state is an abstract representation of the environment (not “world state”)
  - the state space is discrete
- Initial state
- Successor function
- Goal test
- Path cost

Search Problem

- State space
- Initial state:
  - usually the current state
  - sometimes one or several hypothetical states (“what if …”)
- Successor function
- Goal test
- Path cost

Search Problem

- State space
- Initial state
- Successor function:
  - [state → subset of states]
  - an abstract representation of the possible actions (discrete)
- Goal test
- Path cost

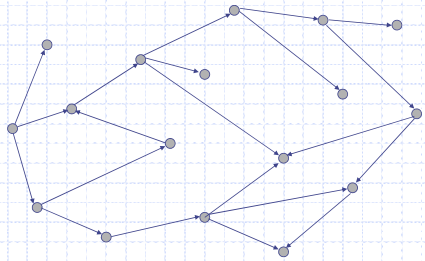
Search Problem

- State space
- Initial state
- Successor function
- Goal test:
  - usually a condition
  - sometimes the description of a state
- Path cost

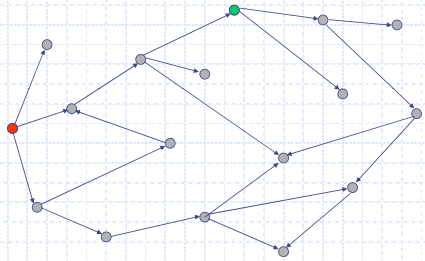
Search Problem

- State space
- Initial state
- Successor function
- Goal test
- Path cost:
  - [path → positive number]
  - usually, path cost = sum of step costs
  - e.g., number of moves of the empty tile

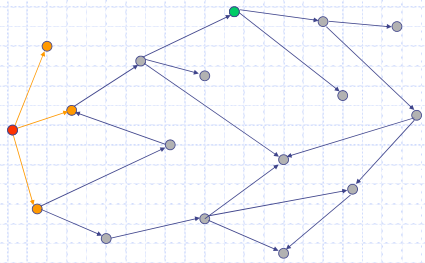
Search of State Space



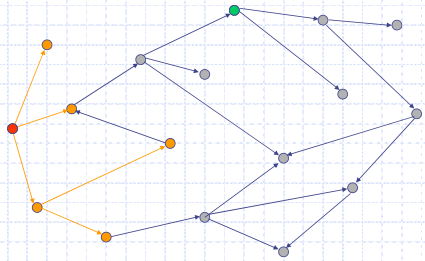
Search of State Space



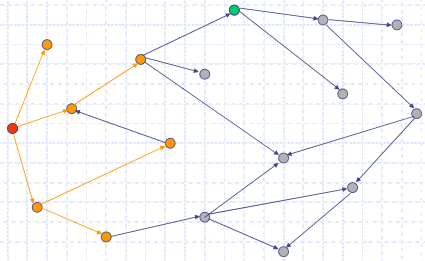
Search of State Space



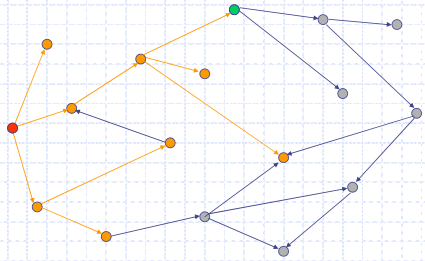
Search of State Space



Search of State Space



Search of State Space



→ search tree

Simple Agent Algorithm

Problem-Solving-Agent

1. initial-state ← sense/read state
2. goal ← select/read goal
3. successor ← select/read action models
4. problem ← (initial-state, goal, successor)
5. solution ← search(problem)
6. perform(solution)

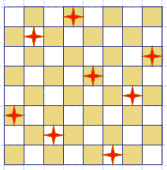
An Old Idea: The Labyrinth and the Ariadne Thread

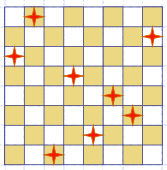
Theseus, a Greek hero, came to Crete to slay the Minotaur, a monster who lived in a Labyrinth. Ariadne gave Theseus a ball of yarn which he unwound as he entered the Labyrinth. After killing the Minotaur, Theseus traced the thread back to the entrance of the Labyrinth, rejoined Ariadne, and successfully escaped Crete.



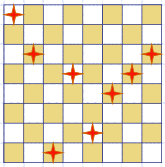
Example: 8-queens

Place 8 queens in a chessboard so that no two queens are in the same row, column, or diagonal.


A solution


Not a solution

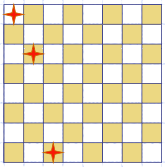
Example: 8-queens



Formulation #1:
- States: any arrangement of 0 to 8 queens on the board
- Initial state: 0 queens on the board
- Successor function: add a queen in any square
- Goal test: 8 queens on the board, none attacked

→ 64^8 states with 8 queens

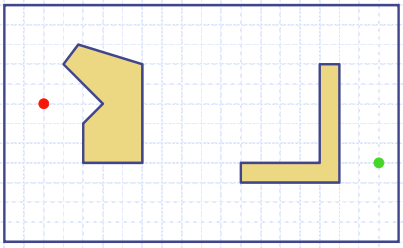
Example: 8-queens



→ 2,057 states

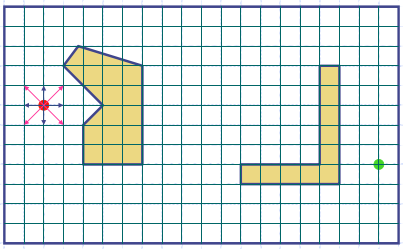
Formulation #2:
- States: any arrangement of k = 0 to 8 queens in the k leftmost columns with none attacked
- Initial state: 0 queens on the board
- Successor function: add a queen to any square in the leftmost empty column such that it is not attacked by any other queen
- Goal test: 8 queens on the board

Example: Robot navigation



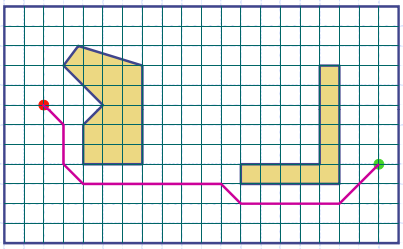
What is the state space?

Example: Robot navigation #1

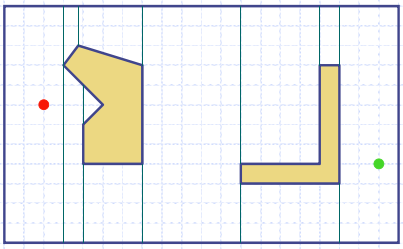


Cost of one horizontal/vertical step = 1
Cost of one diagonal step = $\sqrt{2}$

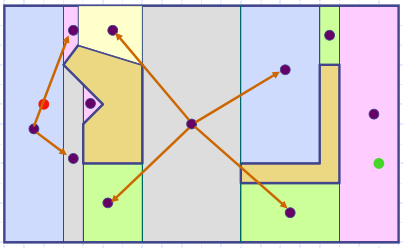
Example: Robot navigation #1



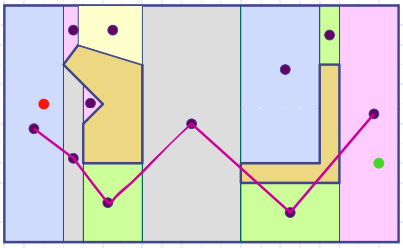
Example: Robot navigation #2



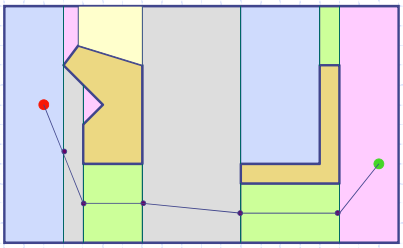
Example: Robot navigation #2



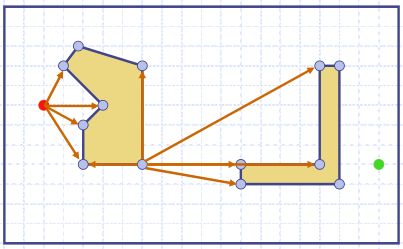
Example: Robot navigation #2



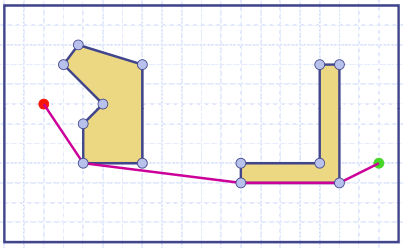
Example: Robot navigation #2



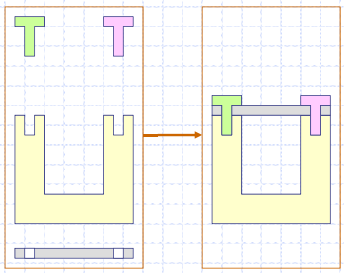
Example: Robot navigation #3



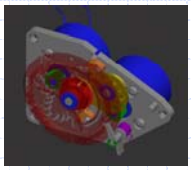
Example: Robot navigation #3



Example: Assembly Planning

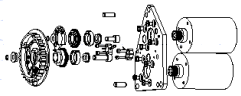


Example: Assembly Planning



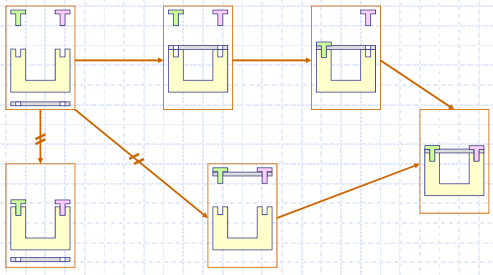
Example: Assembly Planning

- State: Collection of sub-assemblies
- Initial state: All sub-assemblies are individual parts



- Goal state: Complete assembly
- Successor function: Merge two subassemblies (check for collision)
- Cost function: Longest sequence of assembly operation

Example: Assembly Planning



Assumptions in Basic Search

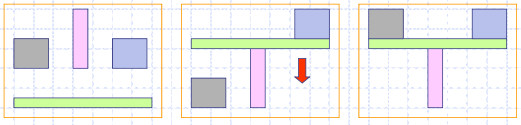
- The environment is static
- The environment is discretizable
- The environment is observable
- The actions are deterministic

Search Problem Formulation

- Real-world environment → Abstraction

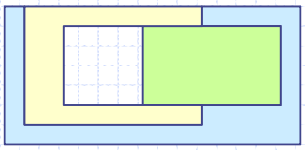
Search Problem Formulation

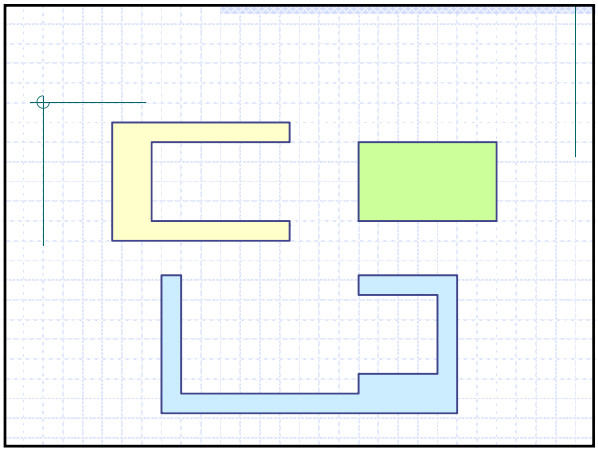
- Real-world environment → Abstraction
  - Validity:
    - Can the solution be executed?

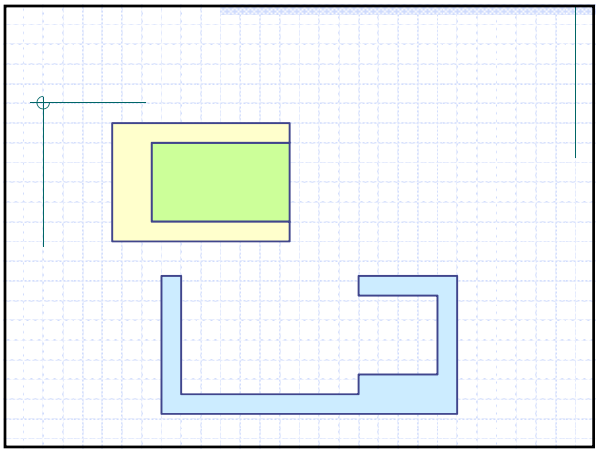


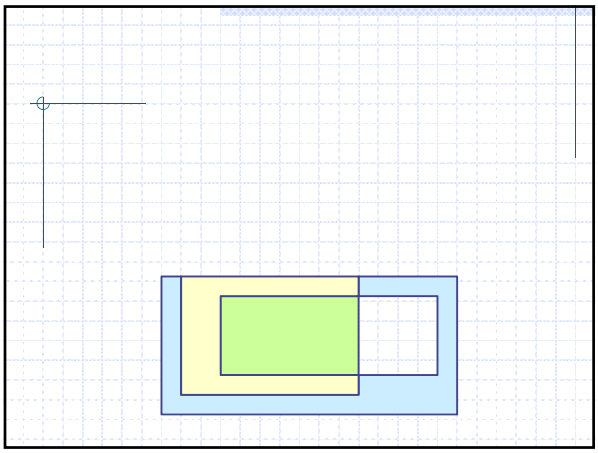
Search Problem Formulation

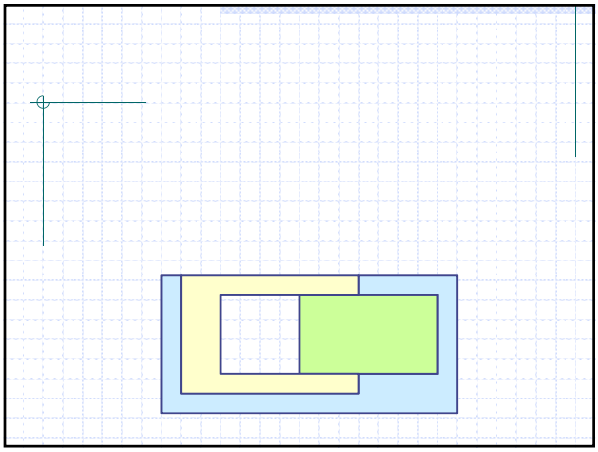
- Real-world environment → Abstraction
  - Validity:
    - Can the solution be executed?
    - Does the state space contain the solution?











Search Problem Formulation

- Real-world environment → Abstraction
  - Validity:
    - Can the solution be executed?
    - Does the state space contain the solution?
  - Usefulness
    - Is the abstract problem easier than the real-world problem?

Search Problem Formulation

- Real-world environment → Abstraction
  - Validity:
    - Can the solution be executed?
    - Does the state space contain the solution?
  - Usefulness
    - Is the abstract problem easier than the real-world problem?
- Without abstraction an agent would be swamped by the complexity of the real world

Search Problem Variants

- One or several initial states
- One or several goal states
- The solution is the path or a goal node
  - In the 8-puzzle problem, it is the path to a goal node
  - In the 8-queen problem, it is a goal node

Search Problem Variants

- One or several initial states
- One or several goal states
- The solution is the path or a goal node
- Any, or the best, or all solutions

Important Parameters

- Number of states in state space

8-puzzle → 181,440
15-puzzle → .65 x 10^{12}
24-puzzle → .5 x 10^{25}

8-queens → 2,057
100-queens → 10^{52}

There exist techniques to solve N-queens problems efficiently!

Stating a problem as a search problem is not always a good idea!

Important Parameters

- Number of states in state space
- Distribution of goal states
- Size of memory needed to store a state

Important Parameters

- Number of states in state space
- Distribution of goal states
- Size of memory needed to store a state
- Running time of the successor function

Applications

- Route finding: airline travel, networks
- Pipe routing, VLSI routing
- Pharmaceutical drug design
- Robot motion planning
- Video games

Summary

- Problem-solving agent
- State space, successor function, search
- Examples: 8-puzzle, 8-queens, route finding, robot navigation, assembly planning
- Assumptions of basic search
- Important parameters