CSE 473: Artificial Intelligence
Autumn 2016

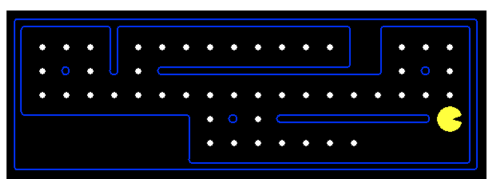
Local Search

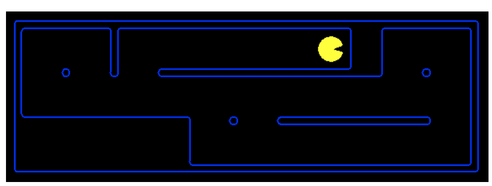
Dan Weld

With slides from
Dan Klein, Stuart Russell, Andrew Moore, Luke Zettlemoyer

Goal Based Agents

- Plan ahead
- Ask "what if"
- Decisions based on (hypothesized) consequences of actions
- Must have a model of how the world evolves in response to actions
- Act on how the world WOULD BE





Types of Environments

- **Fully observable** *vs.* partially observable
- **Single agent** *vs.* multiagent
- **Deterministic** *vs.* stochastic
- **Episodic** *vs.* sequential
- **Discrete** *vs.* continuous

Search thru a Problem Space (aka State Space)

- Input:
  - Set of states
  - Operators *[and costs]*
  - Start state
  - Goal state *[or test]*

Functions: States → States

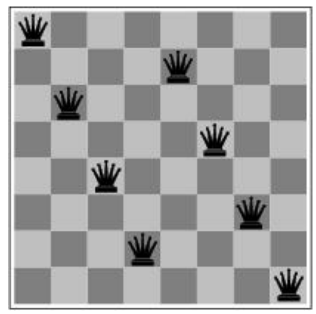
Aka Actions & "Successor Function"

- Output:
  - Path: start ⇒ a state satisfying goal test
    [May require shortest path]
    [Sometimes just need a state that passes test]

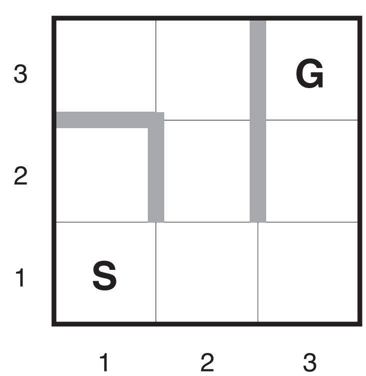
N Queens Problem

Place N queens so they don't attack each other (same row, same col, same diagonal)

- States
  Chess board with
  0 or more queens
- Operators
  Add a queen
- Initial
  No queens
- Goal
  N queens



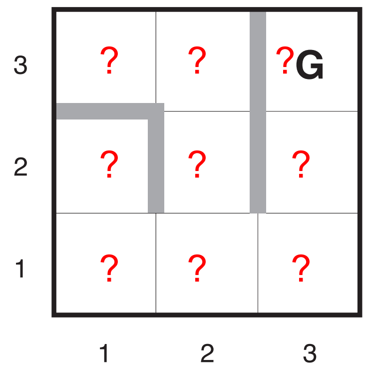
Search thru State Space

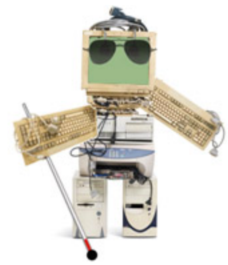


6

What if Robot is Blind?

Moving into wall → noop



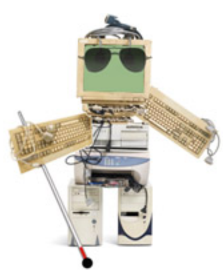


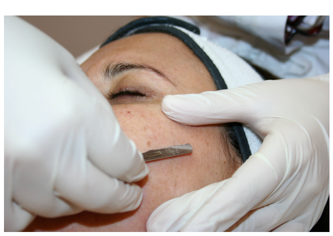
"Conformant Planning"

[Has a talking compass – knows which way is N]

7

Conformant Planning



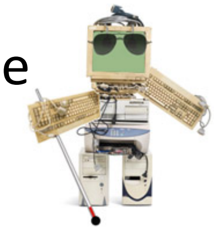


Sterilizing surgical gear

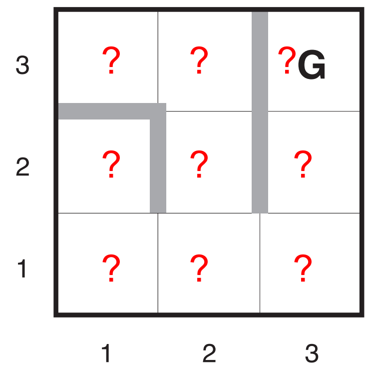
Bowl feeder

8

Search thru State Space



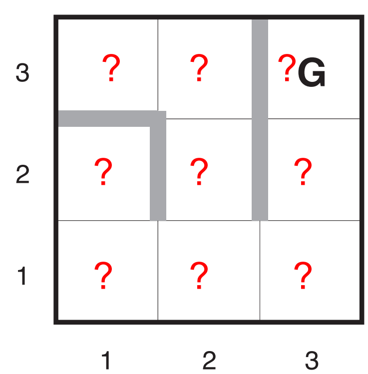
- States
  - SETS of states
  - “Belief state”
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal state(s)



9

Soln: R, D, D, R, R, U, U

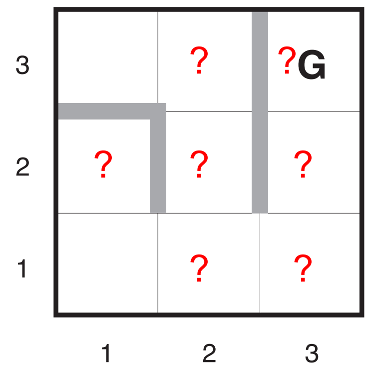
- States
  - SETS of states
  - “Belief state”
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



10

Move Right

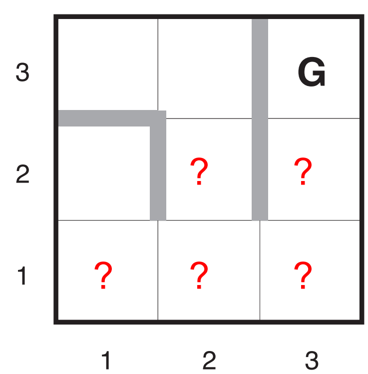
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



11

Move Down

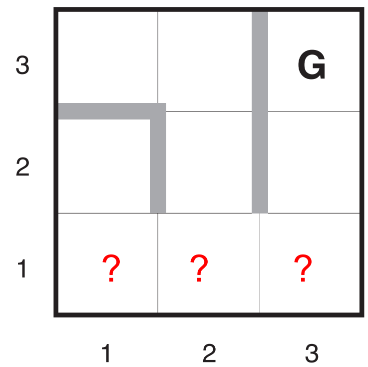
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



12

Move Down

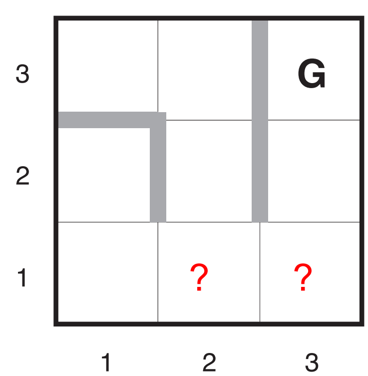
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



13

Move Right

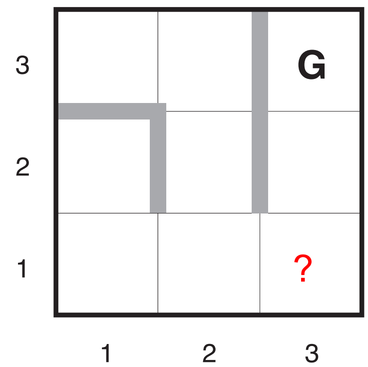
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



14

Move Right

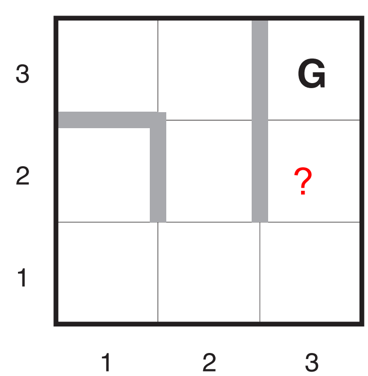
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



15

Move Up

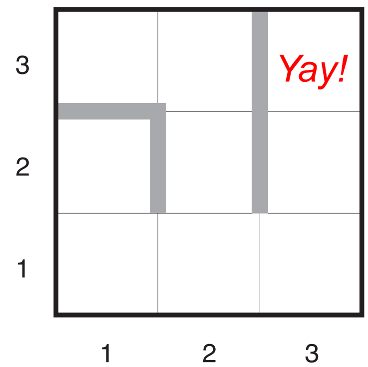
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



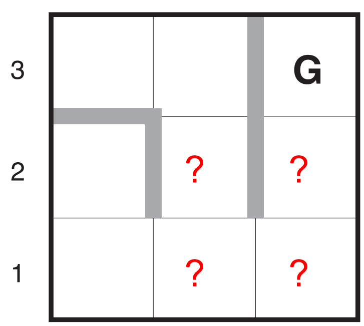
16

Move Up

- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



17



- States
  - SETS of states
  - "Belief state"
- Goal State
  - Set of just goal state(s)

Heuristics?

Relaxed Problem?
- What if it weren't blind?
- Max # moves from any state in belief state

Also… (admissable?)
- Number of states in belief state

18

Previous Search Methods

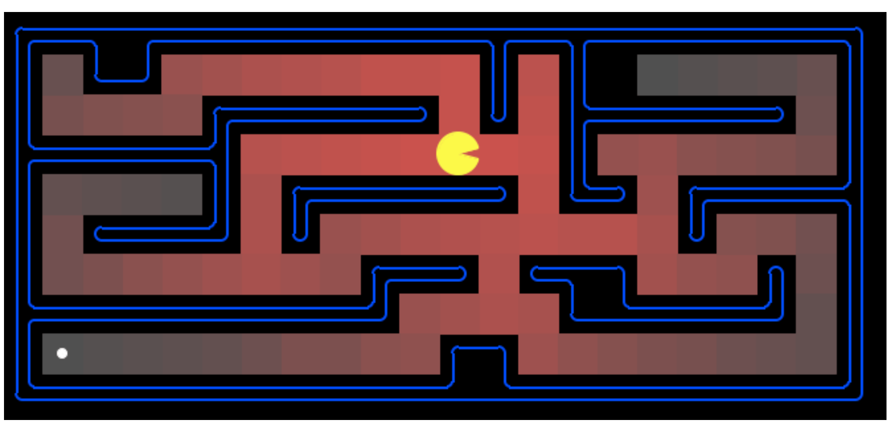
Systematic
- Blind Search
  - Depth first search
  - Breadth first search
  - Iterative deepening search
  - Uniform cost search
- Informed Search
  - Best First
  - A*
  - Beam Search
  - Hill Climbing

Heuristic = Estimate of solution cost

Local (Randomized)
Constraint Satisfaction (Factored)

Which Algorithm?

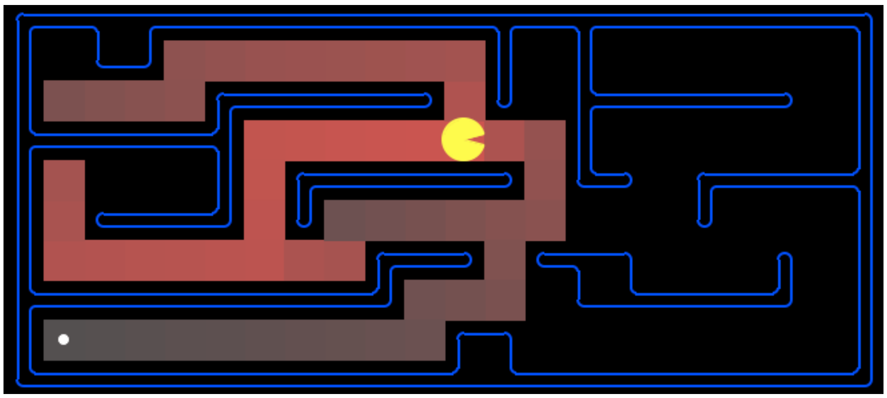
- Uniform cost search (UCS):



20

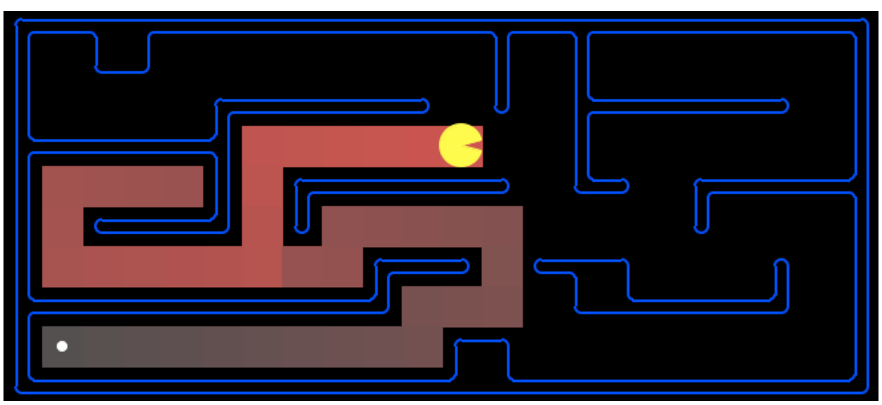
Which Algorithm?

- A*, Manhattan Heuristic:



Which Algorithm?

- Best First / Greedy, Manhattan Heuristic:



Demo

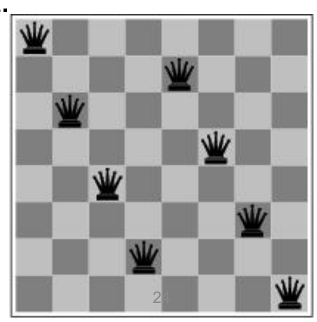
http://qiao.github.io/PathFinding.js/visual/

SUGGESTED BY **Fernando Centurion**

23

Goal State vs Path

- Previously: Search to find best path to goal
  - Systematic exploration of search space.
- Today: a state is solution to problem
  - for some problems path is irrelevant.
  - E.g., 8-queens
- Different algorithms can be used
  - Systematic Search
  - Local Search
  - Constraint Satisfaction



Local search algorithms

- State space = set of "complete" configurations
- Find configuration satisfying constraints,
  - e.g., all n-queens on board, no attacks
- In such cases, we can use local search algorithms
- keep a single "current" state, try to improve it.
- Very memory efficient
  - *duh* - only remember current state

Goal Satisfaction

Constraint satisfaction
reach the goal node
guided by heuristic fn

Optimization

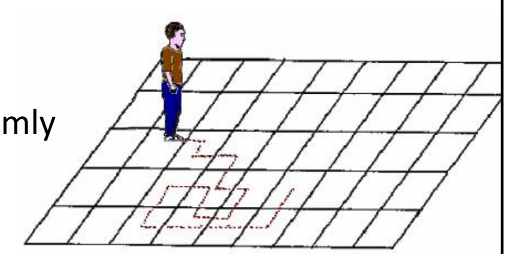
Constraint Optimization
optimize(objective fn)

You can go back and forth between the two problems
Typically in the same complexity class

Local Search and Optimization

- Local search
  - Keep track of single current state
  - Move only to "neighboring" state
    Defined by operators
  - Ignore previous states, path taken
- Advantages:
  - Use very little memory
  - Can often find reasonable solutions in large or infinite (continuous) state spaces.
- "Pure optimization" problems
  - All states have an objective function
  - Goal is to find state with max (or min) objective value
  - Does not quite fit into path-cost/goal-state formulation
  - Local search can do quite well on these problems. 27

Trivial Algorithms



- Random Sampling
  - Generate a state randomly
- Random Walk
  - Randomly pick a neighbor of the current state
- Why even mention these?
  - Both algorithms asymptotically complete.
  - http://projecteuclid.org/download/pdf_1/euclid.aop/1176996718 for Random Walk

28

Hill-climbing search

- "a loop that continuously moves towards increasing value"
  - terminates when a peak is reached
  - Aka greedy local search
- Value can be either
  - Objective function value
  - Heuristic function value (minimized)
- Hill climbing does not look ahead of the immediate neighbors
- Can randomly choose among the set of best successors
  - if multiple have the best value
- "climbing Mount Everest in a thick fog with amnesia"

30

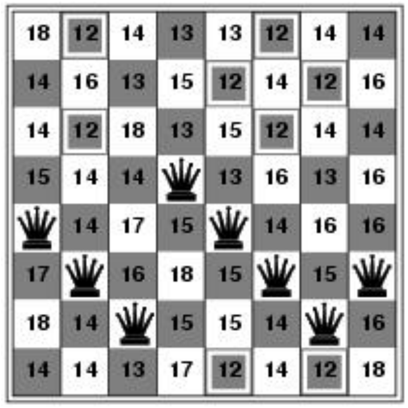
Example: *n*-queens

- Put *n* queens on an *n* x *n* board with no two queens on the same row, column, or diagonal
  - Note different search space… all states have N queens

- Is it a satisfaction problem or optimization?

33

Hill-climbing search: 8-queens problem

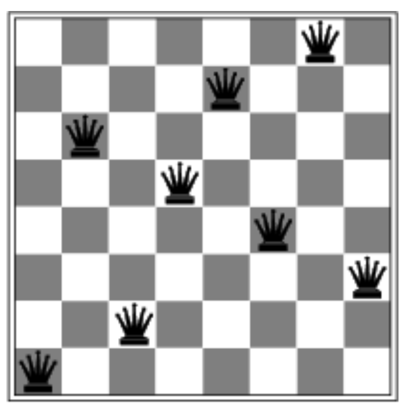


- Need heuristic function
  - Convert to an optimization problem
- *h* = number of ***pairs*** of queens attacking each other
- *h = 17* for the above state

34

Hill-climbing search: 8-queens

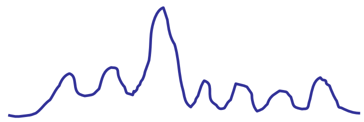
Oooops



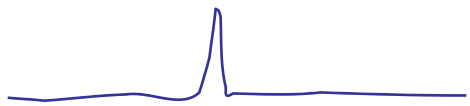
A local minimum with *h = 1*

Hill Climbing Drawbacks

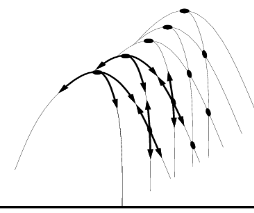
- Local maxima



- Plateaus



- Diagonal ridges



38

Hill Climbing Properties

- Not Complete
- Worst Case Exponential Time
- Simple, O(1) Space & Often Very Fast!

39

Hill-climbing on 8-queens

- Randomly generated 8-queens starting states…
- 14% the time it solves the problem
- 86% of the time it get stuck at a local minimum

- However…
  - Takes only 4 steps on average when it succeeds
  - And 3 on average when it gets stuck
  - (for a state space with 8^8 =~17 million states)

40

Escaping Shoulders: Sideways Move

- If no downhill (uphill) moves, allow sideways moves in hope that algorithm can escape
  - Must limit the number of possible sideways moves to avoid infinite loops
- For 8-queens
  - Allow sideways moves with limit of 100
  - Raises percentage of problems solved from 14 to 94%

  - However….
    - 21 steps for every successful solution
    - 64 for each failure

41

Escaping Local Optima - Enforced Hill Climbing

- Perform breadth first search from a local optima
  - to find the next state with better h function
- Typically,
  - prolonged periods of exhaustive search
  - bridged by relatively quick periods of hill-climbing
- Middle ground b/w local and systematic search

Hill Climbing: stochastic variations

→When the state-space landscape has local minima, any search that moves only in the greedy direction cannot be complete

→Random walk, on the other hand, *is* asymptotically complete

Idea: Combine random walk & greedy hill-climbing

Hill-climbing with random restarts

- If at first you don't succeed, try, try again!
- Different variations
  - For each restart: run until termination vs. run for a fixed time
  - Run a fixed number of restarts or run indefinitely
- Analysis
  - Say each search has probability p of success
    - E.g., for 8-queens, p = 0.14 with no sideways moves
  - Expected number of restarts?

Restarts	0	2	4	8	16	32	64
Success?	14%	36%	53%	74%	92%	99%	99.994%

  - Expected number of steps taken?

Use this algorithm!

Hill-climbing with random walk

- At each step do one of the two
  - Greedy: With prob p move to the neighbor with largest value
  - Random: With prob 1-p move to a random neighbor

Hill-climbing with both

- At each step do one of the three
  - Greedy: move to the neighbor with largest value
  - Random Walk: move to a random neighbor
  - Random Restart: Start over from a new, random state