CSE 573: Artificial Intelligence

Heuristics & Pattern Databases

Dan Weld

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

Outline

- Constructing Heuristics
- Pattern Databases
- Bonus Search Algorithms (Incomplete Ones)

- Reading:
  R&N through Section 3.6 and Sections 4.1 & 4.2

2

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 "Successor Function"

- Output:
  - Path: start ⇒ a state satisfying goal test

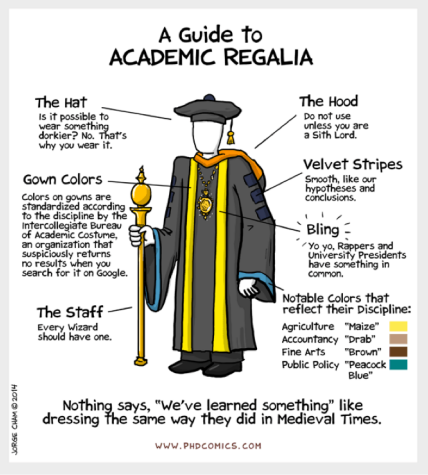
    [May require shortest path]

    [Sometimes just need a state that passes test]

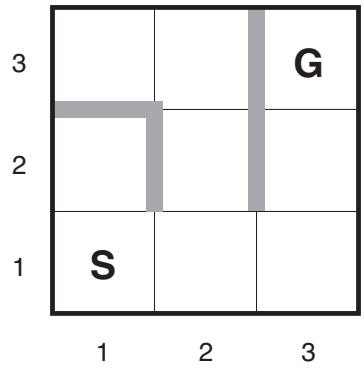
Getting a BS In CSE

Input:

- Set of states
- Operators [costs]
- Start state
- Goal state (test)



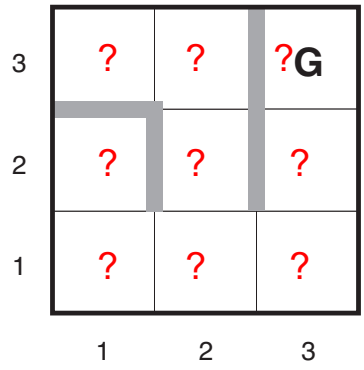
Search thru State Space

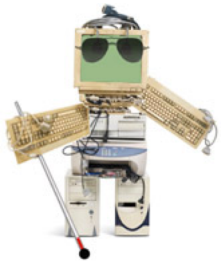


5

What if Robot is Blind?

Moving into wall → noop

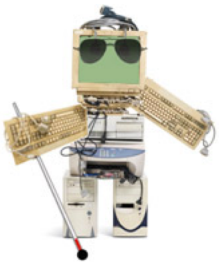




"Conformant Planning"

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

6

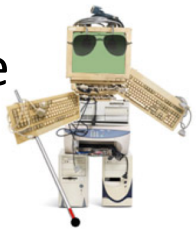


Conformant Planning

Sterilizing surgical gear

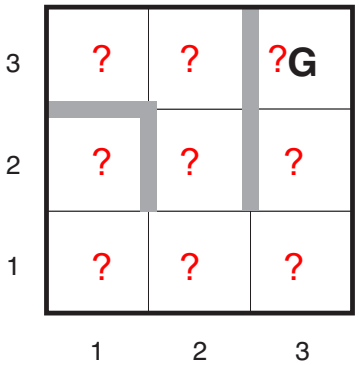
Bowl feeder

7



Search thru State Space

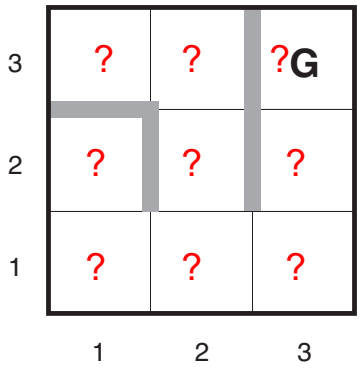
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal state(s)



8

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

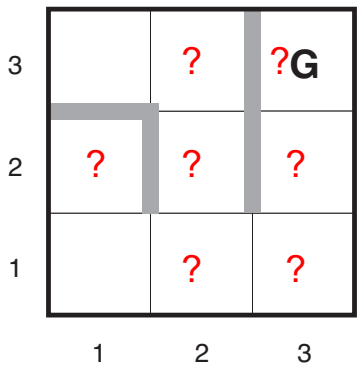
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



9

Move Right

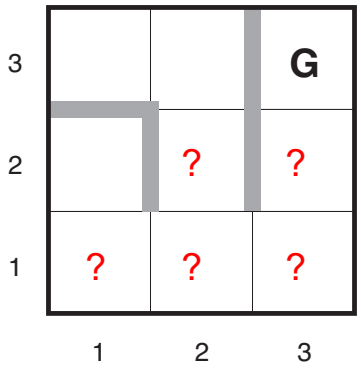
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



10

Move Down

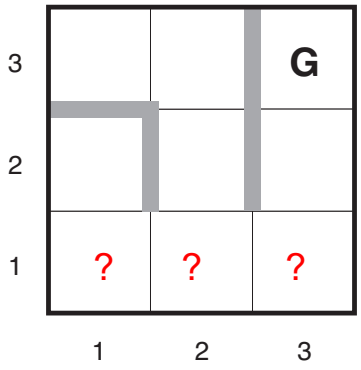
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



11

Move Down

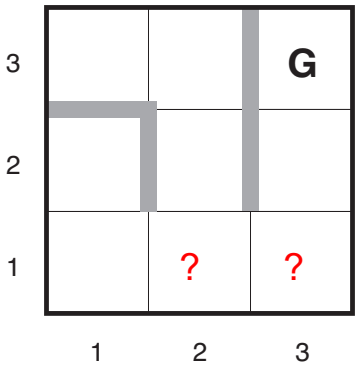
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



12

Move Right

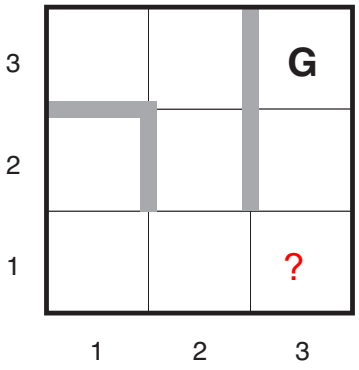
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



13

Move Right

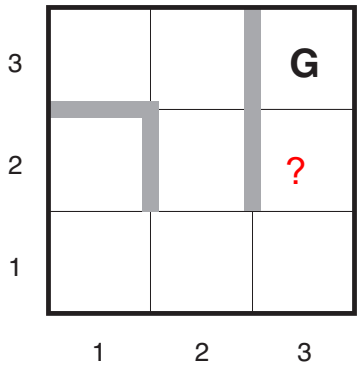
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



14

Move Up

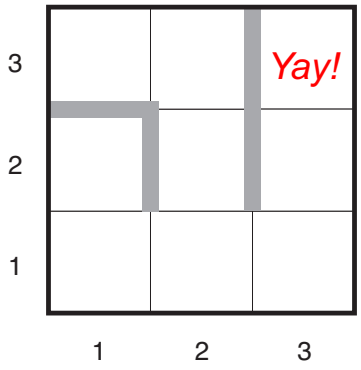
- States
  - SETS of states
  - "Belief state"
- Operators
  - Move actions
- Initial State
  - Set of all states
- Goal State
  - Set of just goal states



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



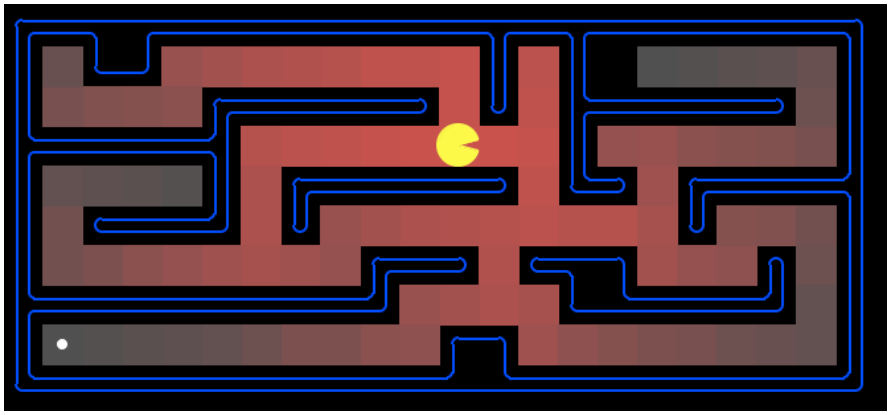
16

Last Week's Methods

- Search Problems
- Uninformed Search Methods
  - Depth-First Search
  - Breadth-First Search
  - Iterative Deepening Search
  - Uniform-Cost Search
- Heuristic Search Methods
  - Uniform Cost = best first with F(n) = G(n) = cost-getting-to(n)
  - Greedy = best first with F(n) = H(n) = heuristic-est-distance(n, goal)
  - A* = best first with F(n) = G(n) + H(n)
  - IDA*

Which Algorithm?

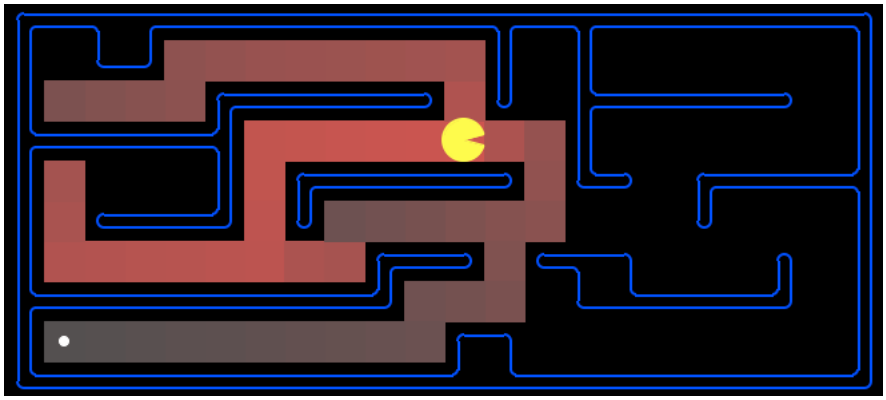
- Uniform cost search (UCS):



18

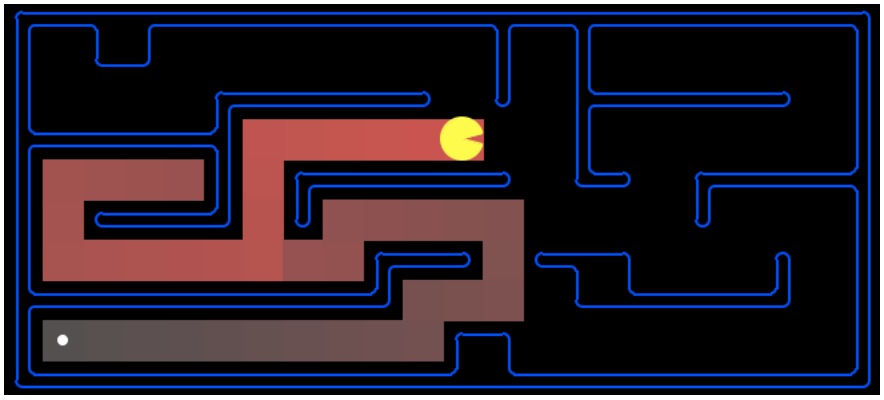
Which Algorithm?

- A*, Manhattan Heuristic:



Which Algorithm?

- Greedy Search, Manhattan Heuristic:



Demo

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

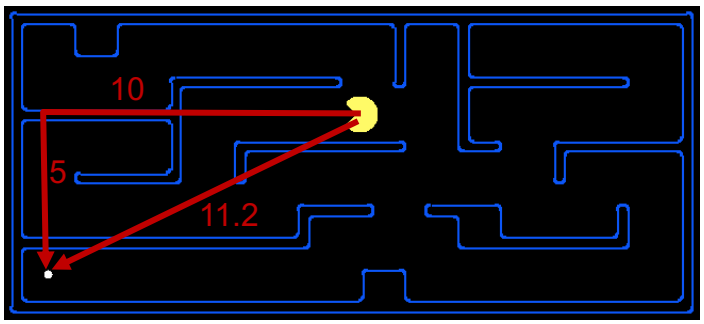
21

Heuristics

How we avoid exponential amounts of search

What is a *Heuristic*?

- An **estimate** of how close a state is to a goal
- Designed for a particular search problem



Admissable Heuristics

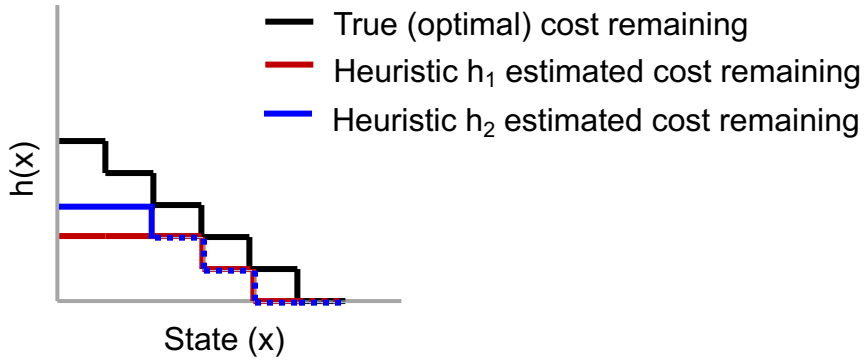
- f(x) = g(x) + h(x)
- g: cost so far
- h: underestimate of remaining costs

Questions
1) Which heuristic is better
2) Where do heuristics come from?

Dominance

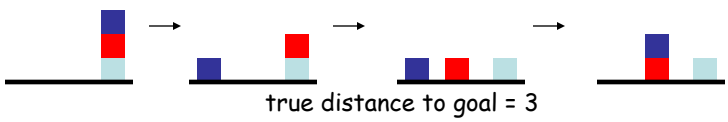
If $h_2(n) \geq h_1(n)$ for all n (both admissible)
then h_2 dominates h_1

h_2 is better - guaranteed never to expand more nodes.



Relaxed Problems

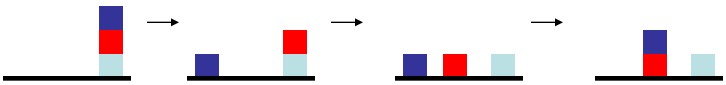
- Derive admissible heuristic from exact cost of a solution to a relaxed version of problem
  - For blocks world, distance = # move operations



- Cost of optimal soln to relaxed problem ≤ cost of optimal soln for real problem

Relaxed Problems

- Derive admissible heuristic from exact cost of a solution to a relaxed version of problem
  - For blocks world, distance = # move operations
  - heuristic = number of misplaced blocks
  - *What is relaxed problem?*



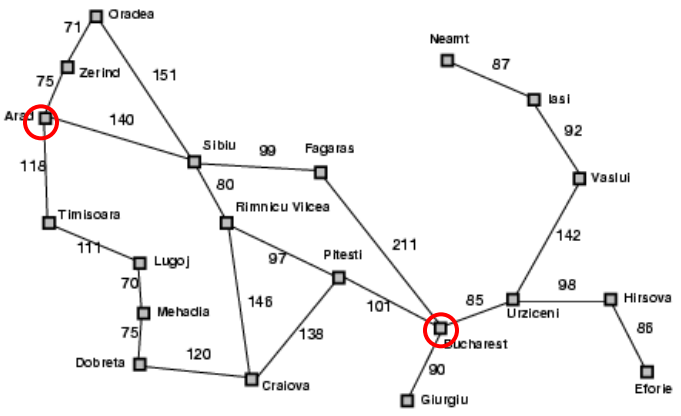
out of place = 2, true distance to goal = 3

- Cost of optimal soln to relaxed problem ≤ cost of optimal soln for real problem

27

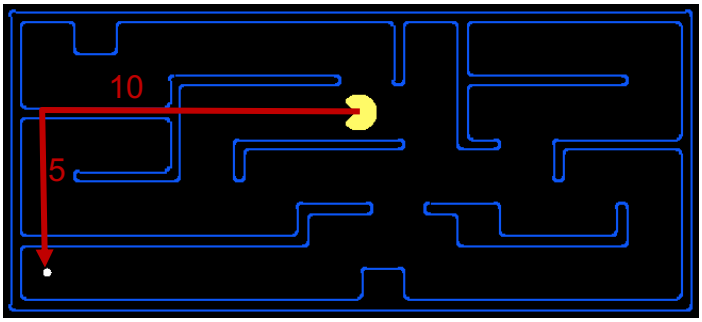
What's being relaxed?

Heuristic = Euclidean distance



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

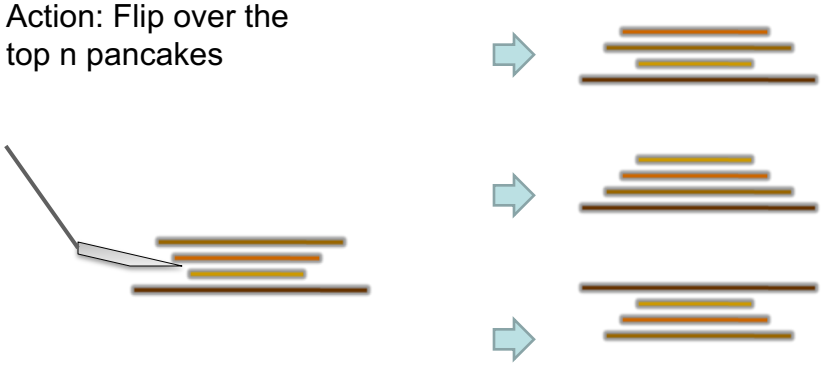
Manhattan Distance



What is Being Relaxed?

Example: Pancake Problem

Action: Flip over the top n pancakes



Cost: ***Number*** of pancakes flipped (not # flip actions)

Example: Pancake Problem

BOUNDS FOR SORTING BY PREFIX REVERSAL

William H. GATES
Microsoft, Albuquerque, New Mexico

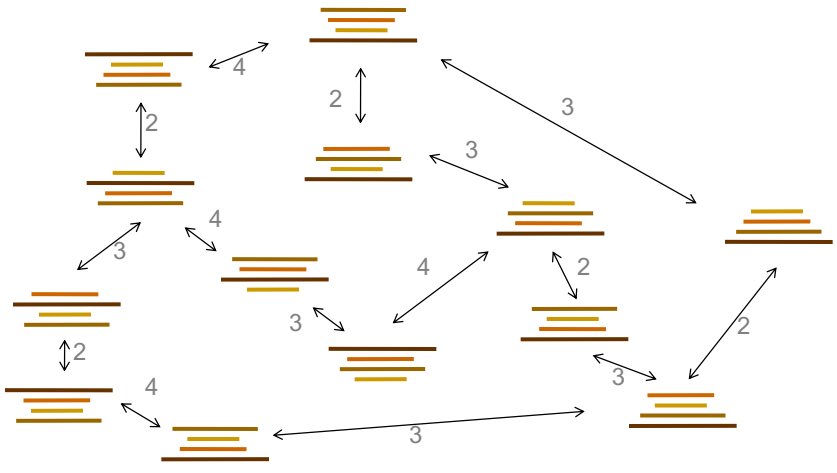
Christos H. PAPADIMITRIOU*†
Department of Electrical Engineering, University of California, Berkeley, CA 94720, U.S.A.

Received 18 January 1978
Revised 28 August 1978

For a permutation σ of the integers from 1 to n, let $f(\sigma)$ be the smallest number of prefix reversals that will transform σ to the identity permutation, and let $f(n)$ be the largest such $f(\sigma)$ for all σ in (the symmetric group) S_n. We show that $f(n) \leq (5n+5)/3$, and that $f(n) \geq 17n/16$ for n a multiple of 16. If, furthermore, each integer is required to participate in an even number of reversed prefixes, the corresponding function $g(n)$ is shown to obey $3n/2 - 1 \leq g(n) \leq 2n+3$.

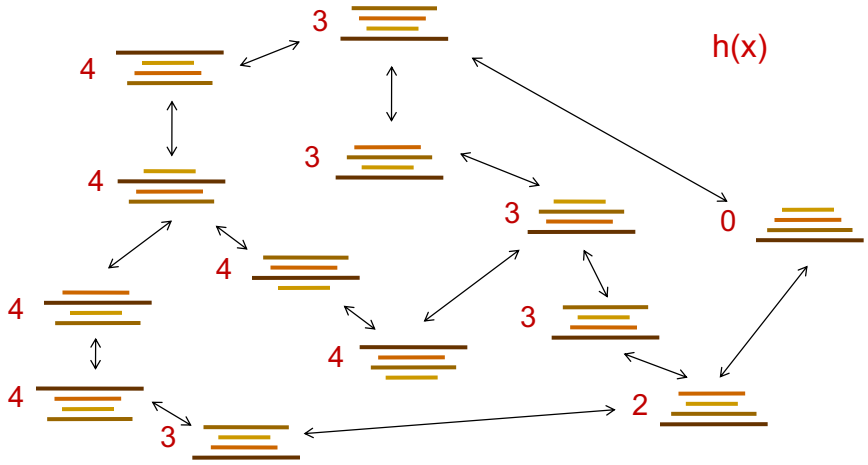
Example: Pancake Problem

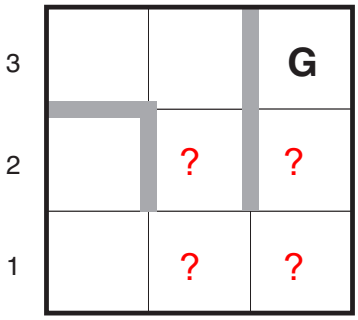
State space graph with costs as weights



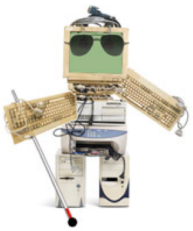
Pancake Heuristic?

Heuristic: the largest pancake that is still out of place





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



Heuristics?

Relaxed Problem?

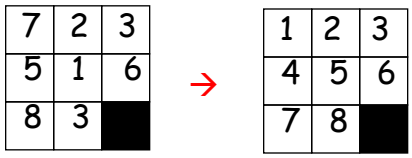
- What if it weren't blind?
- Max # moves from any state in belief state

Also… (admissable?)

- Number of states in belief state

35

Heuristics for eight puzzle



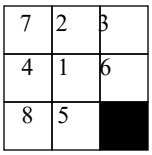
- What can we relax?

h1 = number of tiles in wrong place

h2 = Σ distances of tiles from correct loc

36

Importance of Heuristics



h1 = number of tiles in wrong place

D	A*(h1)
2	6
4	13
6	20
8	39
10	93
12	227
14	539
18	3056
24	39135

37

Importance of Heuristics

7	2	3
4	1	6
8	5	

h1 = number of tiles in wrong place

h2 = Σ distances of tiles from correct loc

D	A*(h1)	A*(h2)
2	6	6
4	13	12
6	20	18
8	39	25
10	93	39
12	227	73
14	539	113
18	3056	363
24	39135	1641

Decrease effective branching factor

38

Starship Raises $25 Million to Roll Out More Delivery Robots

By Chris Albrecht - June 8, 2018 0

Starship Technologies, makers of squat, autonomous wheeled delivery robots, announced yesterday that the company has raised $25 million in additional "seed"

Make your own Admissible Heuristic

- Delivery robot
  - Can carry multiple packages
  - States = location of robot & packages
  - Cost = total distance traveled by robot

40

Need More Power!

Performance of Manhattan Distance Heuristic

- 8 Puzzle < 1 second
- 15 Puzzle 1 minute
- 24 Puzzle 65000 years

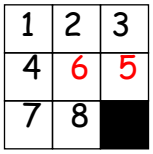
Need even better heuristics!

Adapted from Richard Korf presentation

42

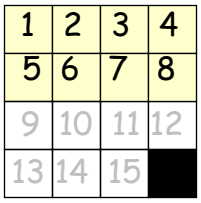
Subgoal Interactions

- Manhattan distance assumes
  - Each tile can be moved independently of others
- Underestimates because
  - Doesn't consider interactions between tiles



Pattern Databases

[Culberson & Schaeffer 1996]



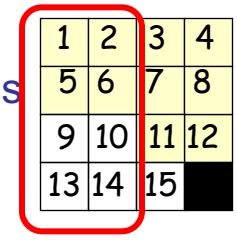
- Pick any subset of tiles
  - E.g., tiles 1-8
- Precompute a table
  - Optimal cost of solving just these tiles
  - For all possible configurations
    - ~ 50 Million in this case
  - Use A* or IDA*
    - State = position of just these tiles (& blank)

Using a Pattern Database

- As each state is generated
  - Use position of chosen tiles as index into DB
  - Use lookup value as heuristic, h(n)
  - Admissible?

Adapted from Richard Korf presentation
45

Combining Multiple Databases



- Can choose another set of tiles
  - Precompute multiple tables
- How combine table values?

- E.g. Optimal solutions to Rubik's cube
  - First found w/ IDA* using pattern DB heuristics
  - Multiple DBs were used (dif cubie subsets)
  - Most problems solved optimally in 1 day
  - Compare with *574,000 years* for IDDFS

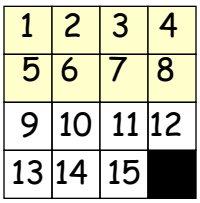
Adapted from Richard Korf presentation
46

Drawbacks of Standard Pattern DBs

- Since we can only take *max*
  - Diminishing returns on additional DBs
- Would like to be able to ***add*** values

Adapted from Richard Korf presentation
47

Disjoint Pattern DBs

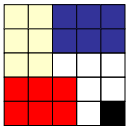


- Partition tiles into disjoint sets
  - For each set, precompute table
    - E.g. 8 tile DB has 519 million entries
    - And 7 tile DB has 58 million
- During search
  - Look up heuristic values for each set
  - *Can add values without overestimating!*
  - Manhattan distance is a special case of this idea where each set is a single tile

Adapted from Richard Korf presentation
48

Performance

- 15 Puzzle: 2000x speedup *vs* Manhattan dist
  - IDA* with the two DBs shown previously solves 15 Puzzles optimally in 30 milliseconds
- 24 Puzzle: 12 million x speedup *vs* Manhattan
  - IDA* can solve random instances in 2 days.
  - Requires 4 DBs as shown
    - Each DB has 128 million entries
  - Without PDBs: 65,000 years



Adapted from Richard Korf presentation

49