Heuristic Search
Heuristics
(Intro to Games)

Andrea Danyluk
February 13, 2017

Announcements

- Programming Assignment 1: Search
  - In progress

Today

- A bit more on A*
- Heuristics
- (Games)

A* Search

- A* search
  - Orders nodes by the sum: f(n) = g(n) + h(n)
    - g(n) is backward cost (to start node)
    - h(n) is forward cost (to closest goal)
  - Optimality requirements
    - Tree search (avoid cycles, but no other notion of "explored" data structure): Heuristic must be **admissible**
      - Never overestimates the cost to the goal

A* Search

- A* search
  - Orders nodes by the sum: f(n) = g(n) + h(n)
    - g(n) is backward cost (to start node)
    - h(n) is forward cost (to closest goal)
  - Optimality requirements
    - Graph Search (keeps track of "explored"): Heuristic must be **consistent**
      - If n^1 is a successor of n generated by action a
        - » h(n) ≤ c(n, a, n^1) + h(n^1)
        - » if an action has cost *c, then taking that action can only cause a drop in heuristic of at most c*

<u>Lemma</u>. If h(n) is consistent, then the values of f(n) along any path are nondecreasing.

Proof:

Let n^1 be a successor of n generated by action a.

Then g(n^1) = g(n) + c(n, a, n^1)

f(n^1) = g(n^1) + h(n^1)

= g(n) + c(n, a, n^1) + h(n^1)

≥ g(n) + h(n), by definition of consistency

Theorem. When A* selects a node for expansion (and marks it "explored"), the optimal path to that node has been found.
Proof: Assume the contrary. Say that a node n^1 has been selected for expansion, but the optimal path to that node has not been found.

If n^1 has been selected for expansion, it must be the case that $f(n^1) \leq f(p)$ for every node p on the optimal path to n that is currently on the frontier.

If the path found is suboptimal, it must be the case that $g(n^1) > g^*(n^1)$, where g^* is the optimal cost of reaching n^1 through p.

Now, we know that our heuristic is consistent, so $f^*(n^1) \geq f(p)$.

But since $h(n^1)$ is the same no matter how you get to n^1, this means $g(n^1) + h(n^1) > g^*(n^1) + h(n^1) = f^*(n^1) \geq f(p)$, but then $f(n^1) > f(p)$, a CONTRADICTION.

Heuristics

- The closer h is to the true cost to the goal, the better A* will perform.
- Better heuristic functions can take more time to compute.
  - Trading off heuristic computation time for search time.
- How to design admissible/consistent heuristics?
- How to make them efficient to compute?

Generating Heuristics

- Calculate the cost to the goal for a relaxed version of the problem
  - Eight-puzzle (or any sliding tile puzzle)
    - Count moves of tiles without taking into account any tiles in the way
    - Count the number of tiles out of place, as if you can lift each tile and place it, without following the rules for movement.
  - Path planning problems
    - Use distance "as the crow flies"

Fifteen Puzzle

- Random 15-puzzle instances first solved optimally by IDA* using Manhattan distance (Korf, 1985)
- Optimal solution lengths average 53 moves
- 400 million nodes generated on average

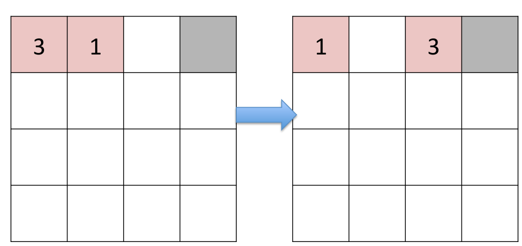
- Twenty-four puzzle: a significantly harder problem. How to solve it?

[This slide and much of the following adapted from Rich Korf]

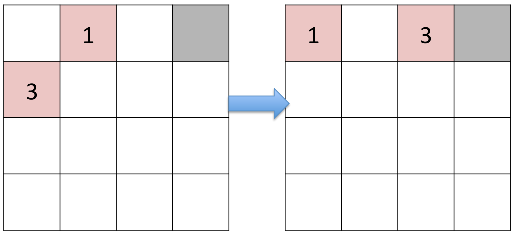
Limitations of Manhattan Distance

- In fact, tiles interfere with each other
- Account for these interactions as efficiently as possible

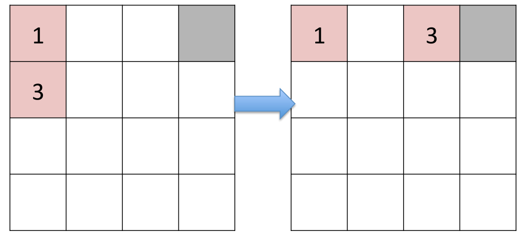
Linear Conflict



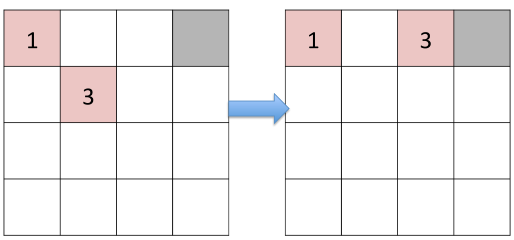
Linear Conflict



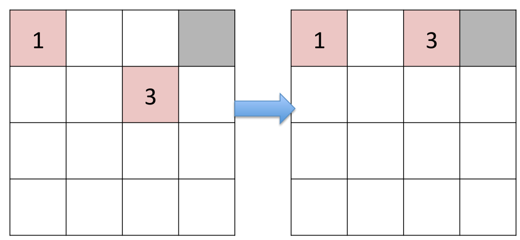
Linear Conflict



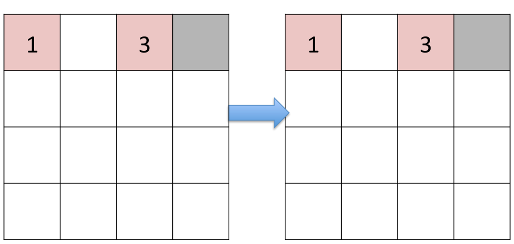
Linear Conflict



Linear Conflict

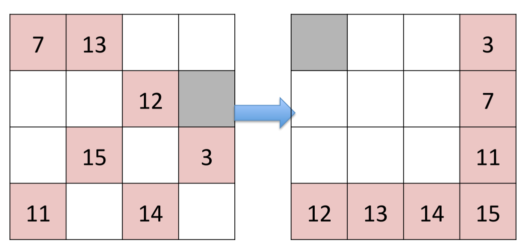


Linear Conflict



Linear Conflict adds two moves to Manhattan Distance (Hanson, Mayer, and Yung, 1991)

More Complex Tile Interactions



MD is 20, but 28 moves are needed. (Counting all tile moves.)

Pattern Database Heuristics

- Culberson and Schaeffer, 1996
- A pattern database is a complete set of such positions, with associated number of moves.
- e.g. a 7-tile pattern database for the Fifteen Puzzle contains 519 million entries.

Additive Pattern Databases

- If no tile belongs to more than one pattern, then we can add their heuristic values.
- Manhattan distance is a special case of this, where each pattern contains a single tile.

Example Additive Databases

	1	2	3
4	5	6	7
8	9	10	11
12	13	15	14

The 7-tile database contains 58 million entries.

The 8-tile database contains 519 million entries
Now counting only the moves of the named tiles.

Computing the Heuristic

5	10	14	7
8	3	6	1
15		12	9
2	11	4	13

→

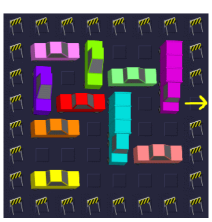
	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

20 moves needed to solve red tiles

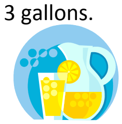
25 moves needed to solve blue tiles

Overall heuristic is sum, or 20+25=45 moves

Heuristic Design Exercise



Heuristic Design Exercises



- Jug-A holds 4 gallons, and Jug-B holds 3 gallons.
- Both initially full.
- Goal: measure exactly 1 gallon.
- Possible actions:
- Fill Jug-A, Filling Jug-B, Empty Jug-A onto the ground, Empty Jug-B onto the ground, Pouring Jug-A into Jug-B (until either Jug-A is empty or Jug-B is full), Pour Jug-B into Jug-A (until either Jug-B is empty or Jug-A is full).

Games

- Planning/problem solving in the presence of an adversary ➔ adversarial search
- Why games?
  - Easy to measure success or failure
  - States and rules are generally easy to specify
  - Interesting and complex
    - Space and time complexity
    - Uncertainty of adversaries' action, rolls of dice, etc.

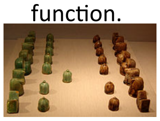
Backgammon

- TDGammon uses depth-2 search + very good evaluation function + reinforcement learning (Gerry Tesauro, IBM)
- World-champion level play
- 1st AI world champion in any game!

[Adapted from CS 188 Berkeley]

Chess

- Deep Blue defeated human world champion Gary Kasparov in a six-game match in 1997.
- Deep Blue examined 200 million positions per second
- Used very sophisticated evaluation function.



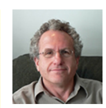
Iranian chess set, 12th century

Claude Shannon

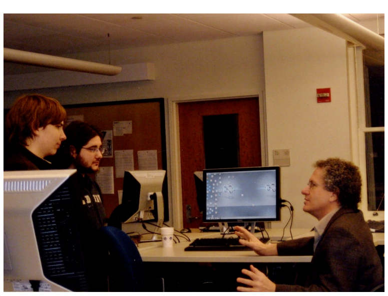
[Adapted from CS 188 Berkeley]

Checkers

- Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994.
- Used an endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 443,748,401,247 positions.
- Checkers is solved!



Othello masters get advice



Go

- AlphaGo became the first program to beat a human professional Go player without handicaps on a full 19x19 board.
- In go, b > 300
- Uses Monte Carlo tree search to select moves.
- Uses knowledge learned from a combination of reinforcement and deep learning.

[Adapted from CS 188 Berkeley]

Poker

- Libratus [Sandholm and Brown, CMU] won $1.7m (in chips) from 4 professional poker players over 20 days in January 2017
- No-limit Texas Hold'em
- Hard because it's a game of imperfect information. Can't see the opponent's hand.
- The "final frontier" in games…