A* Search

Alice Gao

Lecture 4

Readings: R & N 3.5 (esp 3.5.2)

Based on work by K. Leyton-Brown, K. Larson, and P. van Beek

Outline

Learning goals

By the end of the lecture, you should be able to

- Describe motivations for applying informed search algorithms.
- Trace the execution of and implement the A* search algorithm using different heuristic functions.
- Explain why A* is optimally efficient.
- Describe the definition of an admissible heuristic.
- Verify that a heuristic is admissible by showing that it is an optimal solution to a relaxed problem.
- Construct an admissible heuristic for a given search problem.
- Compare different heuristic functions. Give reasons for choosing one heuristic over another.

Why Informed Search?

Assume that these two states are in the frontier.

- How would an uninformed search algorithm choose which one to expand?
- How would humans choose which one to expand?

5	3	
8	7	6
2	4	1

1	2	3
4	5	
7	8	6

Finding the Optimal Solution

- Goal is to find the cheapest path from the start state to a goal state.
- We can make use of two pieces of information.
  - When we are at state n,
  - $g(n)$:
  - $h(n)$:

The Heuristic Function

Definition (search heuristic)

A search heuristic $h(n)$ is an estimate of the cost of the cheapest path from node n to a goal node.

- $h(n)$ is arbitrary, non-negative, and problem-specific.
- If n is a goal node, $h(n) = 0$.
- $h(n)$ must be easy to compute (without search).

Uninformed and Informed Search Algorithms

Treat the frontier as a priority queue ordered by $f(n)$.
What should $f(n)$ be?

- Dijkstra's algorithm (Lowest-Cost-First Search):
  $f(n) = g(n)$.
- Greedy Search:
  $f(n) = h(n)$.
- A* Search:
  $f(n) = g(n) + h(n)$.

The A* Search Algorithm

The frontier is a priority queue ordered by $f(n) = g(n) + h(n)$.
Expand the node with the lowest $f(n)$.

Algorithm 1 Search

```
let the frontier to be an empty list
add initial state to the frontier
while the frontier is not empty do
    remove curr_state from the frontier
    if curr_state is a goal state then
        return curr_state
    end if
    get all the successors of curr_state
    add all the successors to the frontier
end while
return no solution
```

Trace the Execution of A* on the 8-Puzzle

See the notes online.

A* is Optimal

If the heuristic $h(n)$ is admissible, the solution found by A* is optimal.

Definition (admissible heuristic)

A heuristic $h(n)$ is admissible if it is NEVER an OVERestimate of the cost from node n to a goal node. That is, $(\forall n\ (h(n) \leq h^*(n)))$.

A* is Optimally Efficient

Optimal Efficiency: Among all optimal algorithms that start from the same start node and use the same heuristic, A* expands the fewest nodes.

Some Heuristic Functions for 8-Puzzle

- Manhattan Distance Heuristic:
  The sum of the Manhattan distances of the tiles from their goal positions
- Misplaced Tile Heuristic:
  The number of tiles that are NOT in their goal positions

Both heuristic functions are admissible.

Constructing an Admissible Heuristic

- Define a relaxed problem by simplifying or removing constraints on the original problem.
- Solve the relaxed problem without search.
- The cost of the optimal solution to the relaxed problem is an admissible heuristic for the original problem.

Constructing an Admissible Heuristic for 8-Puzzle

8-puzzle: A tile can move from square A to square B

- if square A and square B are adjacent, and
- square B is blank.

Which heuristics can we derive from relaxed versions of this problem?

CQ: Constructing an Admissible Heuristic

CQ: Which heuristics can we derive from the following relaxed 8-puzzle problem?

A tile can move from square A to square B
if A and B are adjacent.

(A) The Manhattan distance heuristic

(B) The Misplaced tile heuristic

(C) Another heuristic not described above

CQ: Constructing an Admissible Heuristic

CQ: Which heuristics can we derive from the following relaxed 8-puzzle problem?

A tile can move from square A to square B.

(A) The Manhattan distance heuristic

(B) The Misplaced tile heuristic

(C) Another heuristic not described above

Which Heuristic is Better?

- We want a heuristic to be admissible.
- Prefer a heuristic that is very different for different states.
- Want a heuristic to have higher values (close to h^*).

Dominating Heuristic

Definition (dominating heuristic)

Given heuristics $h_1(n)$ and $h_2(n)$. $h_2(n)$ dominates $h_1(n)$ if

- $(\forall n \ (h_2(n) \geq h_1(n)))$.
- $(\exists n \ (h_2(n) > h_1(n)))$.

Theorem

If $h_2(n)$ dominates $h_1(n)$, A using h_2 will never expand more nodes than A* using h_1.*

CQ: Which Heuristic of 8-puzzle is Better?

CQ: Which of the two heuristics of the 8-puzzle is better?

(A) The Manhattan distance heuristic dominates the Misplaced tile heuristic.

(B) The Misplaced tile heuristic dominates the Manhattan distance heuristic.

(C) I don't know....

Revisiting the learning goals

By the end of the lecture, you should be able to

- Describe motivations for applying informed search algorithms.
- Trace the execution of and implement the A* search algorithm using different heuristic functions.
- Explain why A* is optimally efficient.
- Describe the definition of an admissible heuristic.
- Verify that a heuristic is admissible by showing that it is an optimal solution to a relaxed problem.
- Construct an admissible heuristic for a given search problem.
- Compare different heuristic functions. Give reasons for choosing one heuristic over another.