# hw2

张芷苒 PB21081601

- 4.1
- 4.2
- 4.6
- 4.7

> **4.1** Trace the operation of$A*$ search applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the $f, g$, and $h$ score for each node.

A* search from Lugoj to Bucharest using the straight-line distance heuristic follows these steps:

1. **Start at Lugoj:** Initial (f(Lugoj) = g(Lugoj) + h(Lugoj)), where (g(Lugoj) = 0) and (h(Lugoj)) is the straight-line distance to Bucharest.
2. **Expand Lugoj:** Calculate (f), (g), and (h) for each neighbor (e.g., Timisoara, Mehadia) based on road distances to these cities ((g)) and their straight-line distances to Bucharest ((h)).
3. **Choose the node with the lowest (f(n))** (sum of (g) and (h)) to expand next.
4. **Repeat** expanding nodes with the lowest (f(n)), updating paths and costs as necessary, until Bucharest is reached.

This process involves continually updating and choosing paths based on the lowest estimated total cost ((f(n))) from Lugoj to Bucharest, considering both actual travel distance so far ((g(n))) and the heuristic estimate to the goal ((h(n))).

> **4.2** The heuristic path algorithm is a best-first search in which the objective function is $f(n) = (2-w)g(n) + wh(n)$. For what values of w is this algorithm guaranteed to be optimal? (You may assume that h is admissible.) What kind of search does this perform when $w = 0$? When $w = 1$?When $w = 2$?

The heuristic path algorithm's objective function is defined as (f(n) = (2-w)g(n) + wh(n)), where (g(n)) represents the cost from the start node to node (n), (h(n)) is the heuristic function estimating the cost from (n) to the goal, and (w) is a weight that balances the influence of (g(n)) and (h(n)). The algorithm is guaranteed to be optimal when the heuristic (h(n)) is admissible, meaning it never overestimates the actual cost to reach the goal.

#### Optimal Values of (w)

The algorithm is optimal when (0 \leq w \leq 1). This is because:

- At (w = 1), the formula becomes (f(n) = g(n) + h(n)), the standard A* search, which is optimal given (h(n)) is admissible.
- At (w = 0), the formula emphasizes only the actual cost so far ((g(n))), essentially performing a uniform cost search.

- For values of (w) between 0 and 1, the algorithm smoothly transitions between these two behaviors, maintaining optimality as long as (h(n)) does not overestimate the costs.

**Behavior for Specific (w) Values**

- **When (w=0):** The formula simplifies to (f(n) = 2g(n)). The factor of 2 has no effect on the search order, making it equivalent to a **Uniform Cost Search** which always expands the node with the lowest path cost from the start, regardless of the estimated cost to the goal.
- **When (w=1):** The formula becomes (f(n) = g(n) + h(n)), which is the standard **A* Search**. This balances the cost so far with the estimated cost to the goal, providing an optimal path as long as (h(n)) is admissible.
- **When (w=2):** The formula simplifies to (f(n) = 2h(n)), completely ignoring the cost so far ((g(n))) and doubling the heuristic estimate. This essentially becomes a **Greedy Best-First Search**, focusing solely on the estimated distance to the goal. The multiplication by 2 does not change the order of node expansion but emphasizes that the actual path cost is disregarded.

In summary, the value of (w) shifts the algorithm's behavior from uniform cost search ((w=0)), through A* search ((w=1)), to greedy best-first search ((w=2)), with optimality guaranteed only when (0 \leq w \leq 1), assuming (h) is admissible.

> **4.6** Invent a heuristic function for the 8-puzzle that sometirnes overestimates, and show how it can lead to a suboptimal solution on a particular problem. (You can use a computer to help if you want.) Prove that, if $h$ never overestimates by more than $c$, $A*$ using $h$ returns a solution whose cost exceeds that of the optimal solution by no more than $c$.

For the 8-puzzle, a heuristic function that sometimes overestimates could be one that calculates the total distance of tiles from their goal positions but also adds a constant value to each tile's distance, regardless of its actual position. Let's call this heuristic (h'), and define it as:

[h'(n) = \text{Total Manhattan Distance} + k]

where (k) is a constant value added for each tile, causing (h') to overestimate.

# Example of Suboptimal Solution

Consider an instance of the 8-puzzle where a tile is one move away from its goal position. The optimal solution cost is 1. However, if (k > 0), (h') might estimate the cost to be greater than 1, potentially leading A* to explore other paths that are longer than the optimal path, resulting in a suboptimal solution.

# Proof of Bounded Suboptimality

Assume (h) is a heuristic that never overestimates by more than (c). This means for any node (n), the actual cost to reach the goal from (n) ((h^*(n))) minus the estimated cost by (h(n)) is at most (c):

[h^*(n) - h(n) \leq c]

When A *uses (h), it ensures that when a goal node (g) is selected for expansion, the cost of the path found to (g) ((g(g))) does not exceed the optimal path cost ((g^*(g))) by more than (c). This is because A *expands nodes in order of increasing (f) value, and a node (n) will not be expanded if there exists a better path that has an (f) value less than or equal to the (f) value of (n). Thus, if (h) overestimates by at most (c), the cost of the solution found by A* will not exceed that of the optimal solution by more than (c).

In essence, the heuristic (h) ensures that A *remains "close" to optimality, with the maximum deviation from the optimal solution cost bounded by the maximum overestimation (c). This guarantees that the solution's cost found by A* using (h) exceeds the optimal solution cost by no more than (c), even if (h) sometimes overestimates.

> **4.7** Prove that if a heuristic is consistent, it must be admissible. Construct an admissible heuristic that is not consistent.

**Consistent Heuristics are Admissible**

**Proof:**
A heuristic (h) is consistent if for every node (n) and successor (n'), (h(n) \leq c(n, n') + h(n')), where (c(n, n')) is the cost from (n) to (n'). To be admissible, (h(n)) must not overestimate the true cost to the goal.

Given the consistency condition, the cost estimated by (h) from the start to the goal via any path will always be less than or equal to the actual cost, because the estimated cost to the goal decreases at least as fast as the actual cost increases as we move from the start to the goal. Therefore, (h) cannot overestimate the true cost, making it admissible.

**Admissible but Not Consistent Heuristic Example**

**Example:**
Consider a heuristic (h(n)) that estimates the cost to reach the goal based on the straight-line distance in a grid where movement is only allowed horizontally or vertically at a constant cost. (h(n)) is the straight-line (diagonal) distance to the goal, which is admissible because it never overestimates the actual cost of reaching the goal, which involves more steps.

However, it's not consistent. Moving one step horizontally or vertically increases the actual cost by one unit but might decrease the straight-line distance by less than one unit, violating the consistency condition (h(n) \leq c(n, n') + h(n')), where (c(n, n')) is the step cost.