“Why is my heuristic behaving this way?!”

True story of a curious student

A good opportunity to do some revisions on A* and heuristics!

Updated on 8th August 2020

f.maire@qut.edu.au

Found this message in my email inbox

“Me and my assignment partner have been steadily working through the assignment and one night he decided to spend all night trying to get a heuristic working and ***just off of a whiff when he returned the cost he multiplied it by 100****. After he did this each of the puzzle* ***times were halved*** *and we have no idea why,*

…

If you can, can you please explain to us why this is and if we are allowed to keep using it.”

Natural questions

- Assume h is an admissible heuristic
- If we use 5*h instead of h, will A* find a goal?
- If we use 64*h instead of h, will A* find a goal?

What is your gut feeling?

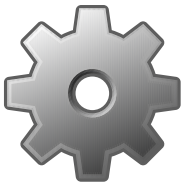
- How does the returned solution (if any) compares with an optimal solution in the worst case?

- Let $h' = 5^* h$ where h admissible

  [5 is arbitrary, our reasoning would work with any other positive constant]

- Let $n' = \underset{x \text{ in Frontier}}{\operatorname{argmin}} \; g(x)+h'(x)$

- Let n be a node of the frontier on an optimal path between the root node and a goal node



Recall

- h' = 5* h where h admissible
- Let n' = argmin g(x)+h'(x)
  x in Frontier
- Let n be a node of the frontier on an optimal path to a goal

---

- g(n') + h'(n') ≤ g(n) + h'(n)
- g(n) + h'(n) = g(n) + 5*h(n)
- g(n) + 5*h(n) ≤ 5*g(n) + 5*h(n)
- 5*g(n) + 5*h(n) ≤ 5*g(n) + 5* h*(n)
- Thefore, g(n') + h'(n') ≤ 5 * optimal_cost

Can you justify each step?

Recall

- h' = 5* h where h admissible
- Let n' = argmin g(x)+h'(x)
  x in Frontier
- Let n be a node of the frontier on an optimal path to a goal

---

- g(n') + h'(n') ≤ g(n) + h'(n) ← By definition of n'
- g(n) + h'(n) = g(n) + 5*h(n) ← By definition of h'
- g(n) + 5*h(n) ≤ 5*g(n) + 5*h(n) ← 1≤5
- 5*g(n) + 5*h(n) ≤ 5*g(n) + 5* h*(n) ← h is admissible
- g(n') + h'(n') ≤ 5 * optimal_cost ← n is on an optimal path

So what?!

- If we use g+5*h instead of g+h to pop a node from the frontier the path returned will cost at most

  5 * optimal_cost

- A* with g+5*h is behaving more like a greedy search and might find a goal more quickly at the cost of a loss of optimality guarantee

Conclusion

- This scenario could happen in any company
  - A software engineer modifies an existing algorithm to tackle a problem
  - Before adopting the modification, any responsible person will want to know about
    - impact on convergence guarantee
    - Impact on performance guarantee

- Software engineers and computer scientists have to wear a mathematician hat sometimes!