Assignment 1

Introduksjon til Kunstig Intelligens

Aditi Ravi Deshpande

September 12, 2025

Contents

1 Task

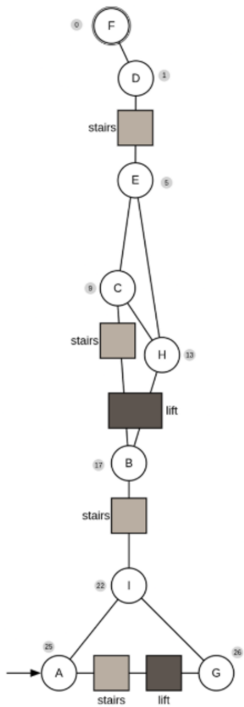

Figure 1: Task

2 Solution

2.1 Setup and Uninformed Search

1. - (A,G) = 4
   - (A,I) = 1
   - (G,I) = 1
   - (I,B) = 2
   - (B,H) = 3
   - (B,C) = 4
   - (H,C) = 1
   - (C,E) = 1
   - (H,E) = 1
   - (E,D) = 2
   - (D,F) = 1

2. (a) BFS: Breadth-First Search. Start node A, goal node F. I use a FIFO *frontier* (queue), a *reached* set to avoid re-enqueuing nodes, and a parent map $P(\cdot)$ to reconstruct the path. Neighbours are generated *in alphabetical order*; ties are broken alphabetically. I will also state here that BFS checks if it is the goal node every time before it expands a node. I have used the pseudocode from [2][p. 95] as inspiration.

- **Initialize:** Frontier = $\langle A \rangle$,
  Reached = $\{A\}$; Root node.
- **Expand** A**:** neighbours G, I
  Frontier = $\langle G, I \rangle$,
  Reached = $\{A, G, I\}$,
  $P(G) = A$,
  $P(I) = A$.
- **Expand** G**:** neighbours A, I already reached.
  Frontier = $\langle I \rangle$,
  Reached = $\{A, G, I\}$.
- **Expand** I**:** neighbours B.
  Frontier = $\langle B \rangle$,
  Reached = $\{A, G, I, B\}$,
  $P(B) = I$.
- **Expand** B**:** neighbours C, H.
  Frontier = $\langle H, C \rangle$,
  Reached = $\{A, G, I, B, C, H\}$,
  $P(C) = B$.
  $P(H) = B$.
- **Expand** C**:** neighbour E, H already reached.
  Frontier = $\langle E, H \rangle$,
  Reached = $\{A, G, I, B, C, H, E\}$,
  $P(E) = C$.
- **Expand** H**:** neighbours C, E already reached
  Frontier = $\langle E \rangle$,
  Reached = $\{A, G, I, B, C, H, E\}$,

- **Expand E:** neighbour D.
  Frontier = $\langle D \rangle$,
  Reached = $\{A, G, I, B, C, H, E, D\}$,
  $P(D) = E$.
- **Expand D:** neighbour F.
  Frontier = $\langle F \rangle$,
  Reached = $\{A, G, I, B, C, H, E, D, F\}$,
  $P(F) = D$.
- **Expand F:** goal found.

Order of expansion: $A, G, I, B, C, H, E, D, F$.
Found path (via parents): $\Rightarrow A \to I \to B \to C \to E \to D \to F$.
Path cost:

$$(A, I) + (I, B) + (B, C) + (C, E) + (E, D) + (D, F)$$

$$= 1 + 2 + 4 + 1 + 2 + 1 = 11.$$

(b) Depth-first Search: The start and end goals are the same as it is for BFS. However, the frontier is a stack, *i.e.* LIFO. So, when we add to the frontier, we add in reverse order. We still have a reached set and a parent map. The algorithm checks if every new node is a goal before expanding. I have used the pseudocode in [2][p. 91] with the node depth as a feature, which is negative.

- **Initialize:** Frontier = $\langle A \rangle$,
  Reached = $\{A\}$; Root node.
- **Expand A:** neighbours G, I
  Frontier = $\langle I, G \rangle$,
  Reached = $\{A, G, I\}$,
  $P(G) = A$,
  $P(I) = A$.
- **Expand G:** neighbours A, I already reached.
  Frontier = $\langle I \rangle$,
  Reached = $\{A, G, I\}$.
- **Expand I:** neighbours B.
  Frontier = $\langle B \rangle$,
  Reached = $\{A, G, I, B\}$,
  $P(B) = I$.
- **Expand B:** neighbours C, H.
  Frontier = $\langle C, H \rangle$,
  Reached = $\{A, G, I, B, C, H\}$,
  $P(C) = B$.
  $P(H) = B$.
- **Expand C:** neighbour E, H already reached.
  Frontier = $\langle H, E \rangle$,
  Reached = $\{A, G, I, B, C, H, E\}$,
  $P(E) = C$.
- **Expand E:** neighbours D C already reached
  Frontier = $\langle H, D \rangle$,
  Reached = $\{A, G, I, B, C, H, E, D\}$,

- **Expand** D**:** neighbour F.
  Frontier $= \langle H, F \rangle$,
  Reached $= \{A, G, I, B, C, H, E, D, F\}$,
  $P(D) = E$.
- **Expand** F**:** goal found.

Order of expansion: A, G, I, B, C, E, D, F.

Found path (via parents): $\Rightarrow A \to G \to I \to B \to C \to E \to D \to F$.

Path cost:

$$(A, G)+(G, I)+(I, B)+(B, C)+(C, E)+(E, D)+(D, F)$$

$$= 4 + 1 + 2 + 4 + 1 + 2 + 1 = 15.$$

(c) Uniform-cost search (UFS): Also known as Dijkstra's algorithm. I have used the pseudocode in [2][p. 91] with path-cost as the evaluation function.

- **Initialize:** Frontier $= \langle A \rangle$
  Cost $= 0$
- **Expand** A**:** neighbours $G(4), I(1)$. We go to I.
  Fromtier $= \{I, G\}$,
  Reached $= \{A : 0, I : 1, G : 4\}$,
- **Expand** I**:** neighbour A, already reached, $G(2), B(4)$.
  We go to G.
  Frontier $= \langle G, B \rangle$,

Reached $= \{A : 0, I : 1, G : 2, B : 3\}$,

- **Expand** G**:** neighbours A, I all reached.
  Frontier $= \langle B \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3\}$,

- **Expand** B**:** neighbours $C(7), H(6)$. Go to H.
  Frontier $= \langle H, C \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3, H : 6, C : 7\}$,

- **Expand** H**:** neighbours $C(7), E(7)$.
  Frontier $= \langle C, E \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3, H : 6, E : 7, C : 7\}$,

- **Expand** C**:** neighbours $E(8)$, not better than E(7), so we ignore.
  Frontier $= \langle E \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3, H : 6, E : 7, C : 7\}$

- **Expand** E**:** neighbours $D(9)$.
  Frontier $= \langle D \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3, H : 6, E : 7, C : 7, D : 9\}$,

- **Expand D:** neighbours $F(10)$.
  Frontier $= \langle F \rangle$,
  Reached $= \{A : 0, I : 1, G : 2, B : 3, H : 6, E : 7, C : 7, D : 9, F : 10\}$,

- **Expand F:** Goal.

Found path: $\Rightarrow A \to I \to G \to B \to H \to E \to C \to D \to F$.
Path cost:

$$(A, I) + (I, G + (I, B) + (B, H) + (H, E)$$

$$+(H, C) + (E, D) + (D, F)$$

$$= 1 + 1 + 2 + 3 + 1 + 1 + 2 + 1 = 12$$

2.2 Part Two: Informed search

3. (a) **Greedy Best - First Search algorithm; Heuristic function of n:** $h(n)$ equals the (unrelated to cost) shortest-path distance to the goal that is F. [2][p. 103]
   Values: $h(F)$=0, $h(D)$=1, $h(E)$=5,
   $h(C)$=9 $h(H)$=13, $h(B)$=17, $h(I)$=22, $h(A)$=25 $h(G)$=26.

   - **Initialize:** Frontier $= \langle (A, h{=}25) \rangle$, Reached $= \{A\}$.

- **Expand A:** neighbours $G(h{=}26),\ I(h{=}22)$
  Frontier $= \langle (I, 22),\ (G, 26) \rangle$
- **Expand I:** neighbours $A(25),\ B(17),\ G(26)$
  Frontier $= \langle (B, 17),\ (G, 26) \rangle$
- **Expand B:** neighbours $C(9),\ H(13),\ I(22)$
  Frontier $= \langle (C, 9),\ (H, 13),\ (G, 26) \rangle$
- **Expand C:** neighbours $B(17),\ H(13),\ E(5)$
  Frontier $= \langle (E, 5),\ (H, 13),\ (G, 26) \rangle$
- **Expand E:** neighbours $C(9),\ H(13),\ D(1)$
  Frontier $= \langle (D, 1),\ (H, 13),\ (G, 26) \rangle$
- **Expand D:** neighbours $E(5),\ F(0)$
  Frontier $= \langle (F, 0),\ (H, 13),\ (G, 26) \rangle$
- **Expand F:** goal found.

Order of expansion: A, I, B, C, E, D, F.
Path cost :

$$(A, I) + (I, B) + (B, C) + (C, E) + (E, D) + (D, F)$$

$$= 1 + 2 + 4 + 1 + 2 + 1 = 11$$

(b) A* search algorithm: In A* each node n is prioritized by

$$f(n) = c(n) + h(n),$$

where $c(n)$ is the path cost from the start to n, and $h(n)$ is the heuristic estimate, same as for Greedy Best-First search. [1][p. 9] I have not shown the neighbours of the nodes that have already been visited.

- **Initialize:** Frontier $= \langle A \rangle$,
- **Expand A:** neighbours $G = f(30)$, $I = f(23)$
  Frontier $= \langle (I, 23), (G, 30) \rangle$
- **Expand I:** neighbours $B(20)$, $G(27)$
  Frontier $= \langle (B, 20), (G, 27) \rangle$
- **Expand B:** neighbours $C(16)$, $H(19)$
  Frontier $= \langle (C, 16), (H, 19), (G, 27) \rangle$
- **Expand C:** neighbours $H(21)$, $E(13)$
  Frontier $= \langle (E, 13), (H, 21), (G, 27) \rangle$
- **Expand E:** neighbours $H(22)$, $D(11)$
  Frontier $= \langle (D, 11), (H, 22), (G, 27) \rangle$
- **Expand D:** neighbours $F(11)$
  Frontier $= \langle (F, 11), (H, 22), (G, 27) \rangle$
- **Expand F:** goal found.

Order of expansion: A, I, B, C, E, D, F.
Path cost :

$$(A, I) + (I, B) + (B, C) + (C, E) + (E, D) + (D, F)$$

$$= 1 + 2 + 4 + 1 + 2 + 1 = 11$$

2.3 A* Search, Admissibility and Consistency

Why did A* not behave as expected? In our implementation, we used a graph-search version of A* with a closed set that keeps only *one value* per node. This means if a node n is first discovered via a suboptimal path **i.e.**, that it has a larger heuristic value ($h(n)$), we discard any later discovery of n that might yield a smaller value $h(n)$.

4. Our graph needs the property of consistency. *The book Artificial Intelligence, A Modern Approach* [2][p. 106], states that A heuristic $h(n)$ is consistent if for every node n and all of its successors n' generated by an action a, we have :

$$h(n) \ \leq \ c(n, a, n') + h(n'). \tag{1}$$

   where c is the step cost of taking action a in the node n that leads to the neighbour n', that is, the cost between them.

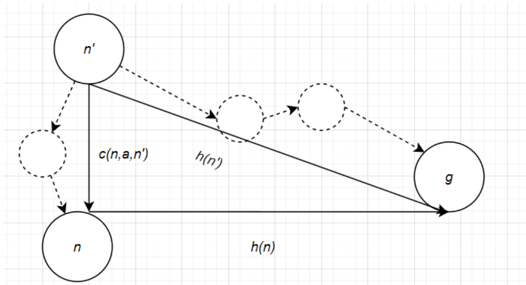

Figure 2: Proving consistency

If we refer to Figure 2, assuming that n is a node, and n' is its neighbour, and the cost of reaching n' from n through action a is *c(n,a,n')*. *h(n)* is the heuristic value of reaching the goal g from n, and similarly *h(n')* is the heuristic value of reaching the goal g from n'. The triangle inequality stipulates that a side of a triangle cannot be longer than the sum of the other two sides. From this, we can state that 1 has to be true.

$$h(n) \ \leq \ c(n, a, n') + h(n').$$

Intuitively, consistency ensures that the estimated cost from n to the goal is **never more** than the cost of stepping to a neighbour n' plus the estimate from there. With consistency, once A* expands a node n, the $h(n)$ found is already optimal.

5. Looking back at figure, let's assume that n is the node C, and n' is its neighbour node E. $h(C) = 9$, $h(E) = 5$, and the cost c is 1. If we add the cost and the heuristic value of E, it does not follow the equation of consistency as :

$$h(C) \ \leq \ c(C, a, E) + h(E).$$

$$9 \ \not\leq \ 1 + 5 = 6$$

   Therefore, we can conclude that the graph is not consistent.

6. A* Search on the simplified graph.
   Updated Heuristic: $h'(F)$=0, $h'(D)$=1, $h'(E)$=3, $h'(C)$=$h'(H)$=4, $h'(B)$=7, $h'(I)$=9, $h'(A)$=$h'(G)$=10.
   I will add the cost depending on the path that the graph takes.

   - **Initialize:** Frontier = $\langle A \rangle$,
     Reached = $\{A : 10\}$.
   - **Expand** A $(G = 14, I = 10)$:
     Frontier = $\langle (I, 10),\ (G, 14) \rangle$;
     Reached = $\{A : 10, I : 10, G : 14\}$.
   - **Expand** I $(G = 12, B = 10)$:
     improve G : (G=12)
     Frontier = $\langle (B, 10),\ (G, 12) \rangle$;
     Reached = $\{A : 10, I : 10, G : 12, B : 10\}$.
   - **Expand** B $(C = 11, H = 10)$:
     Frontier = $\langle (H, 10),\ (C, 11),\ (G, 12) \rangle$;

Reached $= \{A : 10, I : 10, G : 12, B : 10, H : 10, C : 11\}$.

- **Expand** H $(C = 11, E = 10)$:
Frontier $= \langle (E, 10),\ (C, 11)\ (G, 12)\rangle$;
Reached $= \{A : 10, I : 10, G : 12, B : 10, H : 10, C : 11, E : 10\}$.
- **Expand** E $(C = 11, D : 10)$:
Frontier $= \langle (D, 10),\ (C, 11), \rangle$;
Reached $= \{A : 10, I : 10, G : 12, B : 10, H : 10, C : 11, E : 10, D : 10\}$.
- **Expand** D $(F = 10)$:
Frontier $= \langle (F, 10)\rangle$;
Reached $= \{A : 10, I : 10, G : 12, B : 10, H : 10, C : 11, E : 10, D : 10, F : 10\}$.
- **Expand** F : goal found.

Order of expansion: A, I, B, H, E, D, F. **Total path cost:** $1 + 2 + 3 + 1 + 2 + 1 = 10$.

7. Yes, the updated heuristic function makes the graph admissible, as the graph is consistent. To prove that the graph is consistent, please refer to question 8. As mentioned in the book *Artificial Intelligence, A Modern Approach* [2][p. 106], a heuristic function is automatically admissible if it is consistent.

8. Checking each edge in the graph:

$$
\begin{aligned}
h'(A) &= 10 \leq 1+9 = 10 = c(A,I) + h'(I),\\
h'(A) &= 10 \leq 4+10 = 14 = c(A,G) + h'(G),\\
h'(G) &= 10 \leq 1+9 = 10 = c(G,I) + h'(I),\\
h'(I) &= 9 \leq 2+7 = 9 = c(I,B) + h'(B),\\
h'(B) &= 7 \leq 3+4 = 7 = c(B,H) + h'(H),\\
h'(B) &= 7 \leq 4+4 = 8 = c(B,C) + h'(C),\\
h'(H) &= 4 \leq 1+4 = 5 = c(H,C) + h'(C),\\
h'(H) &= 4 \leq 1+3 = 4 = c(H,E) + h'(E),\\
h'(C) &= 4 \leq 2+3 = 5 = c(C,E) + h'(E),\\
h'(E) &= 3 \leq 2+1 = 3 = c(E,D) + h'(D),\\
h'(D) &= 1 \leq 1+0 = 1 = c(D,F) + h'(F).
\end{aligned}
$$

As the consistency equation holds for every node and its neighbour, we can safely say that the heuristic is consistent and automatically admissible.

References

[1] Xavier F. C. Sánchez Díaz. A star search and search in complex environments. Lecture presentation, Course on Introduction to Artificial Intelligence, NTNU, 2025. Available from the course page.

[2] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition, 2020.