FAI Homework Assignment #1

B09902136

March 21, 2023

Problem 1

(a) For depth-first search, the order of state expansion is

$$S, A, B, C, D, G\,.$$

It returns the final path

$$S \to A \to B \to C \to D \to G\,.$$

(b) For breadth-first search, the order of state expansion is

$$S, A, B, C, D, G\,.$$

It returns the final path

$$S \to B \to C \to G\,.$$

(c) For uniform cost search, the order of state expansion is

$$S, B, A, C, D, G\,.$$

It returns the final path

$$S \to B \to D \to G\,.$$

(d) For greedy search, the order of state expansion is

$$S, B, D, G\,.$$

It returns the final path

$$S \to B \to D \to G\,.$$

(e) For A* search, the order of state expansion is

$$S, B, D, G\,.$$

It returns the final path

$$S \to B \to D \to G\,.$$

(f) The heuristic in the graph is admissible.

(g) The heuristic in the graph is not consistent. Note that

$$h(S) = 6 > cost(S \to B) + h(B) = 5\,.$$

We can make it consistent by changing the cost of the edge $S \to B$ to 3.

Problem 2

Let this heuristic be h. Then for every state n:

$$0 \leq h(n) \leq \epsilon h^*(n)\,,$$

where $h^*(n)$ is the true cost to the nearest goal. Let G be the goal node returned by this A* tree search, and G' be the optimal goal node. When A* tree search terminates, there must be some ancestor a of G' on the fringe. Since G was expanded before a, we have

$$g(G) = f(G) \leq f(a) = g(a) + h(a) \leq g(a) + \epsilon h^*(a) \leq \epsilon(g(a) + h^*(a)) = \epsilon g(G')\,.$$

In other words, the cost of the path found by this A* search is at most ϵ times that of the optimal path.

Problem 3

(1) We can formulate this problem as a CSP.

- The variables of the CSP and their domains are

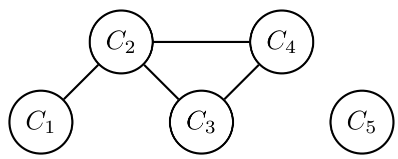
$$
\begin{aligned}
&C_1 \in \{A, C\},\\
&C_2 \in \{A\},\\
&C_3 \in \{B, C\},\\
&C_4 \in \{B, C\},\\
&C_5 \in \{A, B\},
\end{aligned}
$$

  where C_i denotes Class i, and A, B, C denote the instructors.

- The constraints of the CSP are

$$C_1 \neq C_2, \quad C_2 \neq C_3, \quad C_2 \neq C_4, \quad C_3 \neq C_4 \,.$$

(2) Below is the constraint graph of the CSP.



(3) After enforcing all unary constraints and running arc-consistency on the initial graph, we have

- $C_1 \in \{C\}$.
- $C_2 \in \{A\}$.
- $C_3 \in \{B, C\}$.
- $C_4 \in \{B, C\}$.
- $C_5 \in \{A, B\}$.

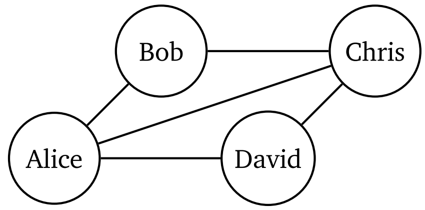
(4) A solution to this CSP is

$$(C_1, C_2, C_3, C_4, C_5) = (C, A, B, C, B)\,.$$

(5) Tree-structured CSPs have no loops in their constraint graphs. Therefore, for a tree-structured CSP, we can greedily assign the nodes in an ordering in which every node's parent precedes it, and find a consistent assignment without any backtracking. Simply put, we can solve tree-structured CSPs more efficiently than general CSPs.

Problem 4

(a) Below is the constraint graph for this CSP.



(b) After running the basic backtracking search, the food assignment should be

$$(\text{Alice}, \text{Bob}, \text{Chris}, \text{David}) = (\text{pizza}, \text{ramen}, \text{pizza}, \text{ramen})\,.$$

(c) After running one iteration of forward checking and assign "pizza" to Alice,

- For Bob, the value "pizza" will be removed.
- For Chris, the values "quesadillas" and "ramen" will be removed.
- For David, the value "pizza" will be removed.