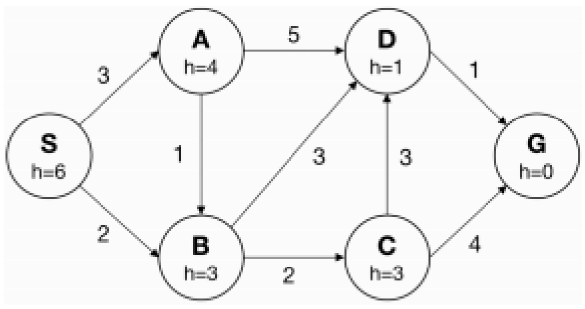
Due: 03/21/2023 23:59

Foundations of Artificial Intelligence: Homework 1

Instructor: Shang-Tse Chen & Yun-Nung Chen

T11902210 張一凡

Problem 1 (14 points)



In the above graph, the start and goal states are S and G, respectively. Write down the order of state expansion and the final path returned by each of the graph search (as oppose to tree search) algorithms below. You can assume ties are resolved alphabetically.

a) Depth-first search.

the order of state expansion: S->A->B->C->D->G
the final path: S->A->B->C->D->G

b) Breadth-first search.

the order of state expansion: S->A->B->D->C->G
the final path: S->A-> D->G

c) Uniform cost search.

the order of state expansion: S->B->A->C->D->G
the final path: S->B-> D->G

d) Greedy search with the heuristic h shown on the graph.

the order of state expansion: S->B->D->G
the final path: S->B-> D->G

e) A^* search with the same heuristic.

the order of state expansion: S->B->D->A->C->G
the final path: S->B-> D->G

(10 points)

f) Is the heuristic in the graph admissible? If not, can you make it admissible by changing the cost of an edge?

Yes, it is admissible.

g) Is the heuristic in the graph consistent? If not, can you make it consistent by changing the cost of an edge?

No, see the following for specific change.
Change the cost of B->D to 4

Problem 2

Suppose the heuristic overestimates the shortest path from any state to the goal by a factor of at most ϵ, where $\epsilon > 1$. Prove that the cost of the path found by A* tree search is at most ϵ times the cost of the optimal path.

Answer:

Denote the cost of the optimal path from the start state to the goal state as C*. Then, according to the A* tree search algorithm, the cost of the path found by A* tree search, denoted as C, is given by:

C = g(goal) = f(goal) - h(goal)

where g(goal) is the actual cost from the start state to the goal state, f(goal) is the estimated total cost from the start state to the goal state, and h(goal) is the heuristic estimate of the cost from the goal state to the goal state.

Since the heuristic estimate is an overestimate, we have:

h(goal) <= C* - g(goal)

Therefore, we can rewrite the above equation as:

C = g(goal) <= f(goal) - h(goal) <= f(goal) - (C* - g(goal)) = f(goal) + g(goal) - C*

Now, let's consider the A* tree search algorithm. At each step, it expands the node with the lowest f-value, which is given by:

f(n) = g(n) + h(n)

where g(n) is the cost of the path from the start state to the current node n, and h(n) is the heuristic estimate of the cost from node n to the goal state.

Suppose that A* tree search expands a set of nodes whose optimal cost is greater than ϵ times the cost of the optimal path. Then, there exists a node n in this set with an f-value of at most ϵC*. Since h(n) is an overestimate, we have:

h(n) <= ϵ(C* - g(n))

Combining this with the equation for f(n), we have:

f(n) = g(n) + h(n) <= g(n) + ϵ(C* - g(n)) = (1-ϵ)g(n) + ϵC*

Since A* tree search expands the node with the lowest f-value, it will not expand any node with an f-value greater than ϵC*. Therefore, the set of nodes expanded by A* tree search has an optimal cost of at most ϵC*, which implies that the cost of the path found by A* tree search is at most ϵ times the cost of the optimal path.

Thus, we have proven that the cost of the path found by A* tree search is at most ϵ times the cost of the optimal path, given that the heuristic overestimates the shortest path from any state to the goal by a factor of at most ϵ.

Problem 3 (10 points)

You are scheduling for 5 classes on the same day taught by 3 instructors. Of course, each instructor can only teach one class at a time.

The classes are:

- Class 1 - Intro to Programming: 8:00-9:00am
- Class 2 - Intro to Artificial Intelligence: 8:30-9:30am
- Class 3 - Natural Language Processing: 9:00-10:00am
- Class 4 - Computer Vision: 9:00-10:00am
- Class 5 - Machine Learning: 10:30-11:30am

The instructors are:

- Instructor A - Can teach Classes 1, 2, and 5.
- Instructor B - Can teach Classes 3, 4, and 5.
- Instructor C - Can teach Classes 1, 3, and 4.

(1) Formulate this problem as a CSP. Describe the variables, domains and constraints.

Variables:
We have eight variables in total:
Class 1: the class "Intro to Programming"
Class 2: the class "Intro to Artificial Intelligence"
Class 3: the class "Natural Language Processing"
Class 4: the class "Computer Vision"
Class 5: the class "Machine Learning"
Instructor A: the instructor who can teach Classes 1, 2, and 5
Instructor B: the instructor who can teach Classes 3, 4, and 5
Instructor C: the instructor who can teach Classes 1, 3, and 4

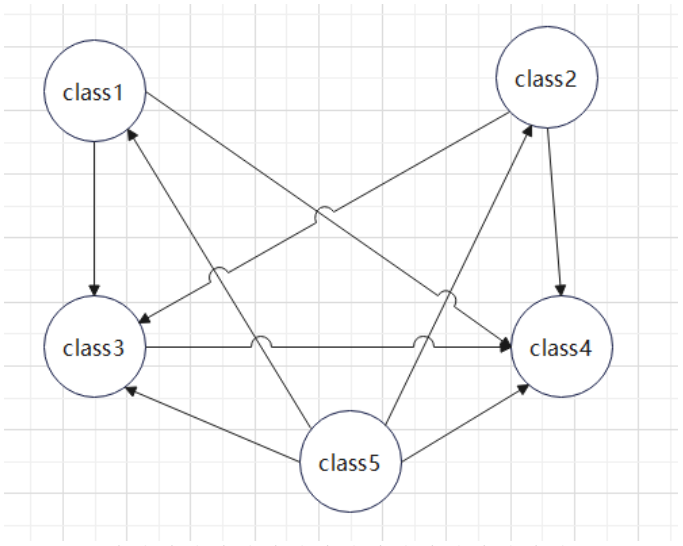
Domains:
The domain for each class variable is the set of available time slots, which are:
8:00-9:00am
8:30-9:30am
9:00-10:00am
10:30-11:30am
The domain for each instructor variable is the set of available classes they can teach. These domains are:
Instructor A: {Class 1, Class 2, Class 5}
Instructor B: {Class 3, Class 4, Class 5}
Instructor C: {Class 1, Class 3, Class 4}

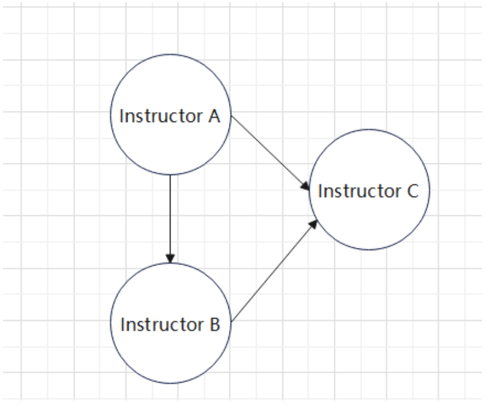
Constraints:
We have two types of constraints in this problem:
Unary constraints: Each class variable has a unary constraint that limits its domain to a single time slot.
Binary constraints: Each pair of variables (class or instructor) that share a time slot has a binary constraint that enforces that they must have different values.
We can represent these constraints as follows:

Unary constraints:
Class 1 $\in$ {8:00-9:00am}
Class 2 $\in$ {8:30-9:30am}
Class 3 $\in$ {9:00-10:00am}
Class 4 $\in$ {9:00-10:00am}
Class 5 $\in$ {10:30-11:30am}
Binary constraints:

Class 1 != Class 3
Class 1 != Class 4
Class 2 != Class 3
Class 2 != Class 4
Class 3 != Class 4
Class 5 != Class 1
Class 5 != Class 2
Class 5 != Class 3
Class 5 != Class 4
Instructor A != Instructor B
Instructor A != Instructor C
Instructor B != Instructor C
These constraints ensure that no two classes are scheduled at the same time, no instructor teaches more than one class at a time, and each class is taught by a different instructor.

(2) Draw the constraint graph associated with your CSP.





In this graph, each variable is represented as a node, and each constraint is represented as a directed edge between two nodes. For example, the edge Class 1 -> Class 3 means that Class 1 and Class 3 share a time slot and therefore cannot have the same value.

(3) Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).

Class 1: {Instructor A}
Class 2: {Instructor A}
Class 3: {Instructor B, Instructor C}
Class 4: {Instructor B, Instructor C}
Class 5: {Instructor A, Instructor B, Instructor C}
Instructor A: {Class 1}, {Class 2}, {Class 5}
Instructor B: {Class 3}, {Class 4}, {Class 5}
Instructor C: {Class 1}, {Class 3}, {Class 4}

(4) Give one solution to this CSP.

Time Slot	Class	Instructor
8:00-9:00am	Intro to Programming (Class 1)	Instructor A
8:30-9:30am	Intro to Artificial Intelligence (Class 2)	Instructor B
9:00-10:00am	Natural Language Processing (Class 3)	Instructor B
9:00-10:00am	Computer Vision (Class 4)	Instructor C
10:30-11:30am	Machine Learning (Class 5)	Instructor A

(5) Your CSP should look nearly tree-structured. Briefly explain (one sentence or less) why we might prefer to solve tree-structures CSPs.

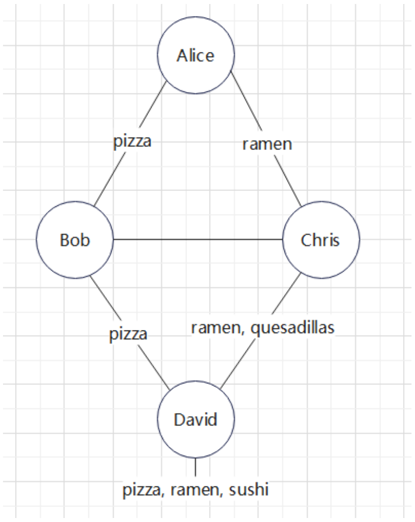
Because they can be solved more efficiently using search algorithms like backtracking, which can exploit the structure of the tree to prune large parts of the search space.

Problem 4 (6 points)

Alice, Bob, Chris, and David are ordering food from pizza, quesadillas, ramen, and sushi. They have some strict preferences:

1. Chris will not order sushi.
2. Alice and Bob want to order different food.
3. Bob will only order pizza or ramen.
4. Alice and Chris want to order the same dish as each other but different from the remaining two people.
5. David will not order quesadillas.

a) Draw the constraint graph for this CSP.



In this graph, the edges indicate the constraints between the nodes. For example, the edge between Alice and Bob represents the constraint that Alice and Bob want to order different food. The edge between Chris and David represents the constraint that David will not order quesadillas. The edge between Alice and Chris represents the constraint that Alice and Chris want to order the same dish as each other but different from the remaining two people.

b) Run the basic backtracking search. Use alphabetical order to both select unassigned variables and iterate over values. Write down the food assignment.

the food assignments are:
1.Alice orders pizza, Bob orders ramen, Chris orders quesadillas, and David orders sushi.
2.Alice orders ramen, Bob orders pizza, Chris orders pizza, and David orders sushi.
3.Alice orders pizza, Bob orders ramen, Chris orders ramen, and David orders pizza.
4.Alice orders ramen, Bob orders pizza, Chris orders pizza, and David orders ramen.
5.Alice orders pizza, Bob orders ramen, Chris orders ramen, and David orders pizza.

b) Assume that no variables have been assigned values yet. When running one iteration of forward checking, which value(s) will be removed for each variable if we assign "pizza" to Alice. Write down "None" if no values will be removed.

The updated domain for each variable after the forward checking step would be:

Alice: pizza

Bob: pizza, ramen

Chris: ramen

David: pizza, ramen, sushi