

CS410: Artificial Intelligence 2020 Fall  
Homework 4: Constraint Satisfaction Problems  
Due date: 23:59:59 (GMT +08:00), December 4, 2020

## 1 Q1

The potluck is coming up and the staff haven't figured out what to bring yet! They've pooled their resources and determined that they can bring some **subset** of the following items.

1. Pho
2. Apricots
3. Frozen Yogurt
4. Fried Rice
5. Apple Pie
6. Animal Crackers

There are five people on the course staff: Taylor, Jonathan, Faraz, Brian, and Alvin. **Each of them** will only bring **one item** to the potluck.

- i If (F)araz brings the same item as someone else, it cannot be (B)rian.
- ii (A)lvin has pho-phobia so he won't bring Pho, but he'll be okay if someone else brings it.
- iii (B)rian is no longer allowed near a stove, so he can only bring items 2, 3, or 6.
- iv (F)araz literally can't even; he won't bring items 2, 4, or 6.
- v (J)onathan was busy, so he didn't see the last third of the list. Therefore, he will only bring item 1, 2, 3, or 4.
- vi (T)aylor will only bring an item that is before an item that (J)onathan brings.
- vii (T)aylor is allergic to animal crackers, so he won't bring item 6. (If someone else brings it, he'll just stay away from that table.)

- viii (F)araz and (J)onathan will only bring items that have the same first letter (e.g. Frozen Yogurt and Fried Rice).
- ix (B)rian will only bring an item that is after an item that (A)lvin brings on the list.
- x (J)onathan and (T)aylor want to be unique; they won't bring the same item as anyone else.

#### Questions

- (a) Which of the listed constraints are unary constraints?
- (b) Rewrite implicit constraint viii. as an explicit constraint.
- (c) Please draw the constraint graph for this CSP. How many edges are there in this constraint graph?

(d) The table below shows the variable domains after all unary constraints have been enforced.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

Following the Minimum Remaining Values heuristic, which variable should we assign first? Break all ties alphabetically.

A ☐ B ☐ F ☐ J ☐ T ☐

(e) To decouple this from the previous question, assume that we choose to assign (F)araz first. In this question, we will choose which value to assign to using the Least Constraining Value method.

To determine the number of remaining values, enforce arc consistency to prune the domains. Then, count the total number of possible assignments (**not** the total number of remaining values). It may help you to enforce arc consistency twice, once before assigning values to (F)araz, and then again after assigning a value.

The domains after enforcing unary constraints are reproduced in each subquestion. **Please delete elements from the grids and write numbers in the blanks.**

(e.i) Assigning F = \_\_\_\_ results in \_\_\_\_ possible assignments.  
The possible assignments are:

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

(e.ii) Assigning F = \_\_\_\_ results in \_\_\_\_ possible assignments.  
The possible assignments are:

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

(e.iii) Assigning F = \_\_\_\_ results in \_\_\_\_ possible assignments.  
The possible assignments are:

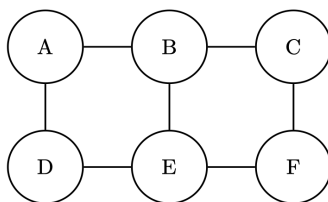
A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

(e.iv) Using the LCV method, which value should we assign to F? If there is a tie, choose the lower number. (e.g. If both 1 and 2 have the same value, then fill 1.)

1 ○      2 ○      3 ○      4 ○      5 ○      6 ○

## 2 Q2.Worst-Case Backtracking

Consider solving the following CSP with standard backtracking search where we enforce arc consistency of all arcs before every variable assignment. Assume every variable in the CSP has a domain size  $d > 1$ .



(a) For each of the variable orderings, mark the variables for which backtracking search (with arc consistency checking) could end up considering more than one different value during the search and **explain why**.

(a.i) Ordering: A, B, C, D, E, F

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

(a.ii) Ordering: B, D, F, E, C, A

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

(b) Now assume that an adversary gets to observe which variable ordering you are using, and after doing so, gets to choose to add one additional binary constraint between any pair of variables in the CSP in order to maximize the number of variables that backtracking could occur in the worst case. For each of the following variable orderings, select which additional binary constraint should the adversary add and **explain why**. Then, mark the variables for which backtracking search (with arc consistency checking) could end up considering more than one different value during the search when solving the modified CSP.

(b.i) Ordering: A, B, C, D, E, F

The adversary should add the additional binary constraint:

AC ☐ AE ☐ AF ☐ BD ☐  
BF ☐ CD ☐ CE ☐ DF ☐

When solving the modified CSP with this ordering, backtracking might occur at the following variable(s):

A ○      B ○      C ○      D ○      E ○      F ○

(b.ii) Ordering: B, D, F, E, C, A

The adversary should add the additional binary constraint:

AC ○      AE ○      AF ○      BD ○  
 BF ○      CD ○      CE ○      DF ○

When solving the modified CSP with this ordering, backtracking might occur at the following variable(s):

A ○      B ○      C ○      D ○      E ○      F ○

### 3 Q3

The final exam of course Artificial Intelligence is coming up, and the course staff has yet to write the test. There are a total of 6 questions on the exam and each question will cover a topic. Here is the format of the exam:

- q1. Search
- q2. Machine Learning Basics
- q3. Logistic Regression
- q4. Neural Networks
- q5. Convolutional Neural Networks
- q6. CSPs

There are 7 people on the course staff: Brad, Donahue, Ferguson, Judy, Kyle, Michael, and Nick. Each of them is responsible to work with the director on one question. (But a question could end up having more than one staff person, or potentially zero staff assigned to it.) However, the staff are pretty quirky and want the following constraints to be satisfied:

- i Donahue (D) will not work on a question together with Judy (J).
- ii Kyle (K) must work on either Search, Machine Learning Basics or Logistic Regression
- iii Michael (M) is very odd, so he can only contribute to an odd-numbered question.
- iv Nick (N) must work on a question that's before Michael (M)'s question.
- v Kyle (K) must work on a question that's before Donahue (D)'s question.
- vi Brad (B) must work on Convolutional Neural Networks.
- vii Judy (J) must work on a question that's after Nick (N)'s question.
- viii If Brad (B) is to work with someone, it cannot be with Nick (N).
- ix Nick (N) cannot work on question 6.
- x Ferguson (F) cannot work on questions 4, 5, or 6
- xi Donahue (D) cannot work on question 5.
- xii Donahue (D) must work on a question before Ferguson (F)'s question.

(a) We will model this problem as a constraint satisfaction problem (CSP). Our variables correspond to each of the staff members, J, F, N, D, M, B, K, and the domains are the questions 1, 2, 3, 4, 5, 6. After applying the unary constraints, what are the resulting domains of each variable? (You can delete elements from the following grid and write the result on the leftside of the grid)

B	1	2	3	4	5	6
D	1	2	3	4	5	6
F	1	2	3	4	5	6
J	1	2	3	4	5	6
K	1	2	3	4	5	6
N	1	2	3	4	5	6
M	1	2	3	4	5	6

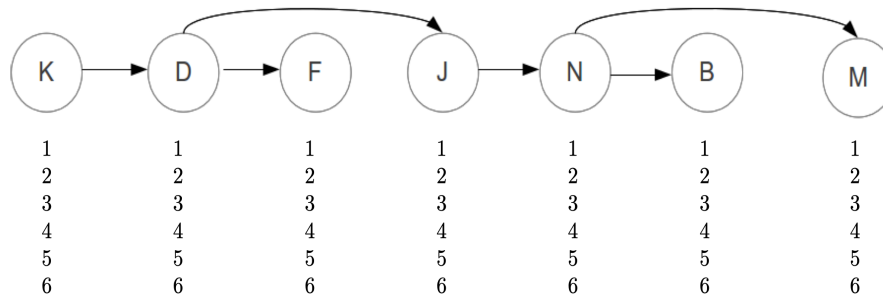
(b) If we apply the Minimum Remaining Value (MRV) heuristic, which variable should be assigned first?

(c) Normally we would now proceed with the variable you found in (b), but to decouple this question from the previous one (and prevent potential errors from propagating), let's proceed with assigning Michael first. For value ordering we use the Least Constraining Value (LCV) heuristic, where we use Forward Checking to compute the number of remaining values in other variables domains. What ordering of values is prescribed by the LCV heuristic? Include your work—i.e., include the resulting filtered domains that are different for the different values.

(d) Realizing this is a tree-structured CSP, we decide not to run backtracking search, and instead use the efficient two-pass algorithm to solve tree-structured CSPs. We will run this two-pass algorithm **after** applying the unary constraints from part (a). Below is the linearized version of the tree-structured CSP graph for you to work with.



(d.i) First Pass: Domain Pruning. Pass from right to left to perform Domain Pruning. Write the values that remain in each domain below each node in the figure above.

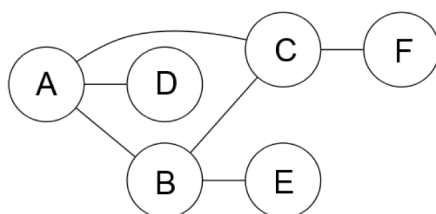


(d.ii) Second Pass: Find Solution. Pass from left to right, assigning values for the solution. If there is more than one possible assignment, choose the highest value.

## 4 Q4

The graph below is a constraint graph for a CSP that has only binary constraints. Initially, no variables have been assigned.

For each of the following scenarios, **mark** all variables for which the specified filtering might result in their domain being changed and **give your reasons**.



(a) A value is assigned to A. Which domains might be changed as a result of running forward checking for A?

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

(b) A value is assigned to A, and then forward checking is run for A. Then a value is assigned to B. Which domains might be changed as a result of running forward checking for B?

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

(c) A value is assigned to A. Which domains might be changed as a result of enforcing arc consistency after this assignment?

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

(d) A value is assigned to A, and then arc consistency is enforced. Then a value is assigned to B. Which domains might be changed as a result of enforcing arc consistency after the assignment to B?

A ☐ B ☐ C ☐ D ☐ E ☐ F ☐

## 5 Q5. Local Search

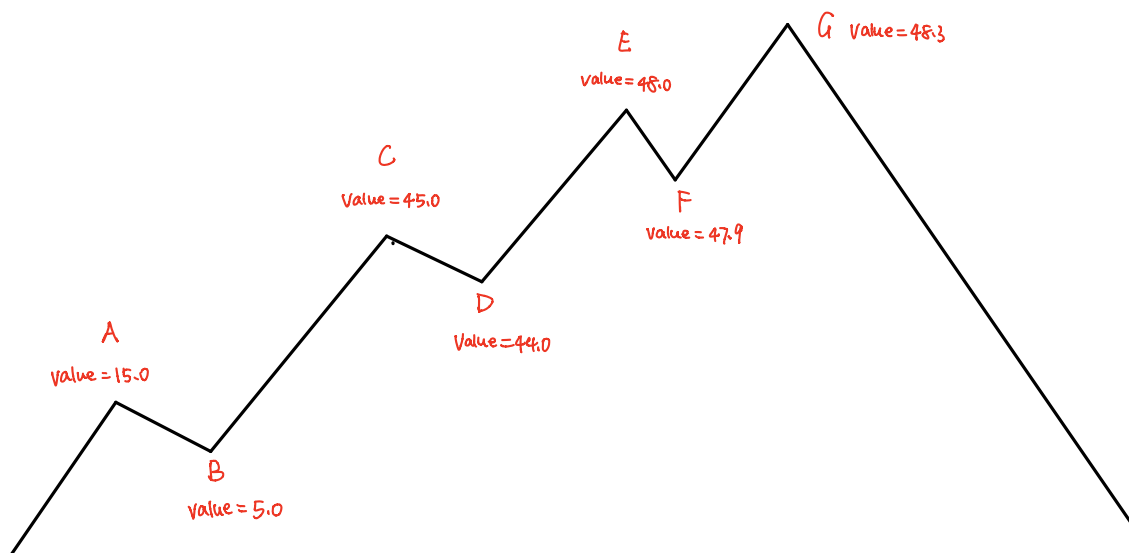
This question asks about Simulated Annealing local search. In the value landscape cartoon below, you will be asked about the probability that various moves will be accepted at different temperatures. Recall that Simulated Annealing always accepts a better move ( $\Delta\text{Value} = \text{Value}[\text{next}] - \text{Value}[\text{current}] > 0$ ); but it accepts a worse move ( $\Delta\text{Value} < 0$ ) only with probability  $e^{\Delta\text{Value}/T}$ , where  $T$  is the current temperature on the temperature schedule.

Please use this temperature schedule (usually, it is a decaying exponential; but it is simplified here):

time(t)	1-100	101-200	201-300
Temperature(T)	10.0	1.0	0.1

The values given have been chosen to follow this table:

$x$	0.0	-0.1	-0.4	-1.0	-4.0	-40.0
$e^x$	1.0	$\approx 0.90$	$\approx 0.67$	$\approx 0.37$	$\approx 0.02$	$\approx 4.0\text{e-}18$



Give your answer to two significant decimal places. The first one is done for you as an example.

(a) (example) You are at Point A and  $t=23$ . The probability you will accept a move  $A \rightarrow B = 0.37$

(b) You are at Point B and  $t=23$ . The probability you will accept a move  $B \rightarrow C =$  \_\_\_\_\_

(c) You are at Point C and  $t=123$ . The probability you will accept a move  $C \rightarrow B =$  \_\_\_\_\_

(d) You are at Point C and  $t=123$ . The probability you will accept a move  $C \rightarrow D =$  \_\_\_\_\_

(e) You are at Point E and  $t=123$ . The probability you will accept a move  $E \rightarrow D =$  \_\_\_\_\_

(f) You are at Point E and  $t=123$ . The probability you will accept a move  $E \rightarrow F =$  \_\_\_\_\_

(g) You are at Point G and  $t=123$ . The probability you will accept a move  $G \rightarrow F =$  \_\_\_\_\_

(h) You are at Point G and  $t=223$ . The probability you will accept a move  $G \rightarrow F =$  \_\_\_\_\_

(i) With a very, very, very long slow annealing schedule, are you more likely, eventually in the long run, to wind up at point A or at point G? (write A or G) \_\_\_\_\_ .