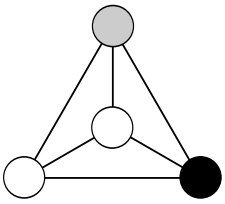
CS 188 Fall 2020 Introduction to Artificial Intelligence

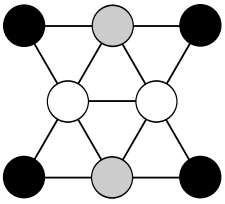
Midterm Review CSPs

Q1. CSPs

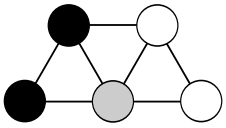
In this question we are considering CSPs for map coloring. Each region on the map is a variable, and their values are chosen from {black, gray, white}. Adjacent regions cannot have the same color. The figures below show the constraint graphs for three CSPs and an assignment for each one. None of the assignments are solutions as each has a pair of adjacent variables that are white. For both parts of this question, let the score of an assignment be the number of satisfied constraints (so a higher score is better).



(1)



(2)



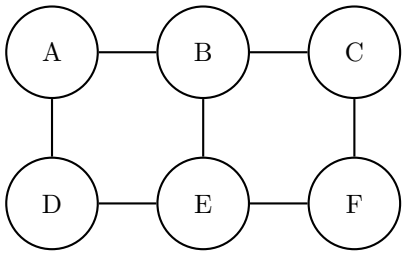
(3)

(a) Consider applying Local Search starting from each of the assignments in the figure above. For each successor function, indicate whether each configuration is a local optimum and whether it is a global optimum (note that the CSPs may not have satisfying assignments).

Successor Function	CSP	Local optimum?		Global Optimum?	
Change a single variable	(1)	Yes	No	Yes	No
	(2)	Yes	No	Yes	No
	(3)	Yes	No	Yes	No
Change a single variable, or a pair of variables	(1)	Yes	No	Yes	No
	(2)	Yes	No	Yes	No
	(3)	Yes	No	Yes	No

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.

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

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(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. 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.

(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

(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

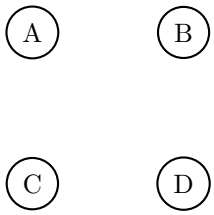
Q3. CSPs

Four people, A, B, C, and D, are all looking to rent space in an apartment building. There are three floors in the building, 1, 2, and 3 (where 1 is the lowest floor and 3 is the highest). Each person must be assigned to some floor, but it's ok if more than one person is living on a floor. We have the following constraints on assignments:

- A and B must not live together on the same floor.
- If A and C live on *the same* floor, they must both be living on floor 2.
- If A and C live on *different* floors, one of them must be living on floor 3.
- D must not live on the same floor as anyone else.
- D must live on a higher floor than C.

We will formulate this as a CSP, where each person has a variable and the variable values are floors.

(a) Draw the edges for the constraint graph representing this problem. Use binary constraints only. You do not need to label the edges.



(b) Suppose we have assigned C = 2. Apply forward checking to the CSP, filling in the boxes next to the values for each variable that are eliminated:

A	☐ 1	☐ 2	☐ 3
B	☐ 1	☐ 2	☐ 3
C		☐ 2	
D	☐ 1	☐ 2	☐ 3

(c) Starting from the original CSP with full domains (i.e. without assigning any variables or doing the forward checking in the previous part), enforce arc consistency for the entire CSP graph, filling in the boxes next to the values that are eliminated for each variable:

A	☐ 1	☐ 2	☐ 3
B	☐ 1	☐ 2	☐ 3
C	☐ 1	☐ 2	☐ 3
D	☐ 1	☐ 2	☐ 3

(d) Suppose that we were running local search with the min-conflicts algorithm for this CSP, and currently have the following variable assignments.

A	3
B	1
C	2
D	3

Which variable would be reassigned, and which value would it be reassigned to? Assume that any ties are broken alphabetically for variables and in numerical order for values.

The variable ◯ A ◯ B ◯ C ◯ D will be assigned the new value ◯ 1 ◯ 2 ◯ 3