CS221 Problem Session Solutions

Week 6

1) **Problem 1: Modeling** Farmer Kim wants to install a set of sprinklers to water all his crops in the most cost-effective manner and has hired you as a consultant. Specifically, he has a rectangular plot of land, which is broken into $W \times H$ cells. For each cell (i, j), let $C_{i,j} \in \{0, 1\}$ denote whether there are crops in that cell that need watering. In each cell (i, j), he can either install $(X_{i,j} = 1)$ or not install $(X_{i,j} = 0)$ a sprinkler. Each sprinkler has a range of R, which means that any cell within Manhattan distance of R gets watered. The maintenance cost of the sprinklers is the sum of the Manhattan distances from each sprinkler to his home located at $(1, 1)$. Recall that the Manhattan distance between (a_1, b_1) and (a_2, b_2) is $|a_1 - a_2| + |b_1 - b_2|$. Naturally, Farmer Kim wants the maintenance cost to be as small as possible given that all crops are watered. See figure below for an example.

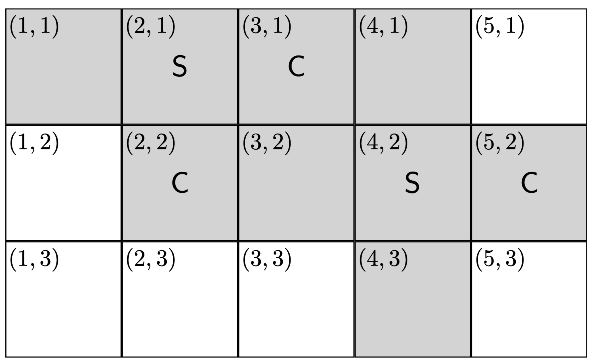


Figure 1: An example of a farm with $W = 5$ and $H = 3$. Each cell (i, j) is marked 'C' if there are crops there that need watering $(C_{i,j} = 1)$. An example of a sprinkler installation is given: a cell (i, j) is marked with 'S' if we are placing a sprinkler there $(X_{i,j} = 1)$. Here, the sprinkler range is $R = 1$, and the cells that are shaded are the ones covered by some sprinkler. In this case, the sprinkler installation is valid (all crops are watered), and the total maintenance cost is $1 + 4 = 5$.

Farmer Kim actually took CS221 years ago, and remembered a few things. He says: "I think this should be formulated as a factor graph. The variables should be $X_{i,j} \in \{0, 1\}$ for each cell (i, j). But here's where my memory gets foggy. What should the factors be?" Let $X = \{X_{i,j}\}$ denote a full assignment to all variables $X_{i,j}$. Your job is to define two types of factors:

- $f_{i,j}$: ensures any crops in (i, j) are watered,
- f_{cost}: encodes the maintenance cost,

so that a maximum weight assignment corresponds to a valid sprinkler installation with minimum maintenance cost.

$f_{i,j}(X) =$

Solution For each cell (i, j), let $f_{i,j}$ encode whether the crops (if they exist) in (i, j) are watered:

$$f_{i,j}(X) = \left[C_{i,j} = 0 \text{ or } \left(\min_{i',j':X_{i',j'}=1} |i' - i| + |j' - j|\right) \leq R\right]. \tag{1}$$

The first part encodes there being no crops in cell (i, j). The second part encodes that if there is a crop, the closest sprinkler should be at most R Manhattan distance away.

$f_{\text{cost}}(X) =$

Solution We define the next factor to the exponentiated negative minimum cost, so that the factor is non-negative and that maximizing the weight corresponds to minimizing the maintenance cost:

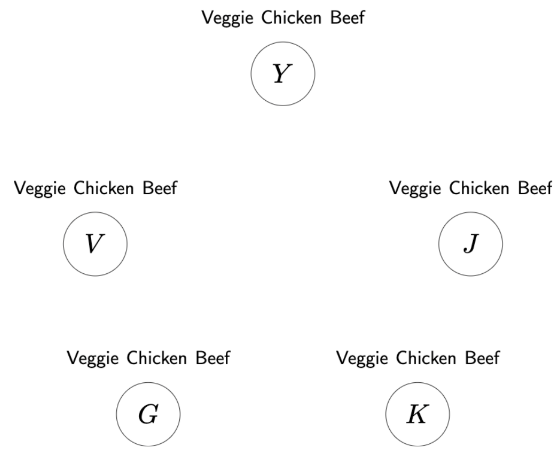
$$f_{\text{cost}}(X) = \exp\left(-\sum_{i',j':X_{i',j'}=1} |i' - 1| + |j' - 1|\right). \tag{2}$$

Any answer where the factor is non-negative and maximizing the weight corresponds to minimizing the maintenance cost is accepted. The answer should also account for the case where there are no sprinklers. Below is another acceptable answer:

$$f_{\text{cost}}(X) = \begin{cases} 1 & \text{if } X_{i',j'} = 0 \text{ for all } i', j' \\ \left(\sum_{i',j':X_{i',j'}=1} |i' - 1| + |j' - 1|\right)^{-1} & \text{otherwise} \end{cases}$$

2) Problem 2: Arc Consistency

(a) You and your friends (Veronica, Jarvis, Gabriela, Kanti) sit around a table like this:



There are three dishes on the menu: the vegetarian deep dish pizza, the chicken quesadilla, and the beef cheeseburger. Each person will order exactly one dish. But what started out as a simple dinner has quickly turned into a logistical nightmare because of all the constraints you and your friends impose upon yourselves:

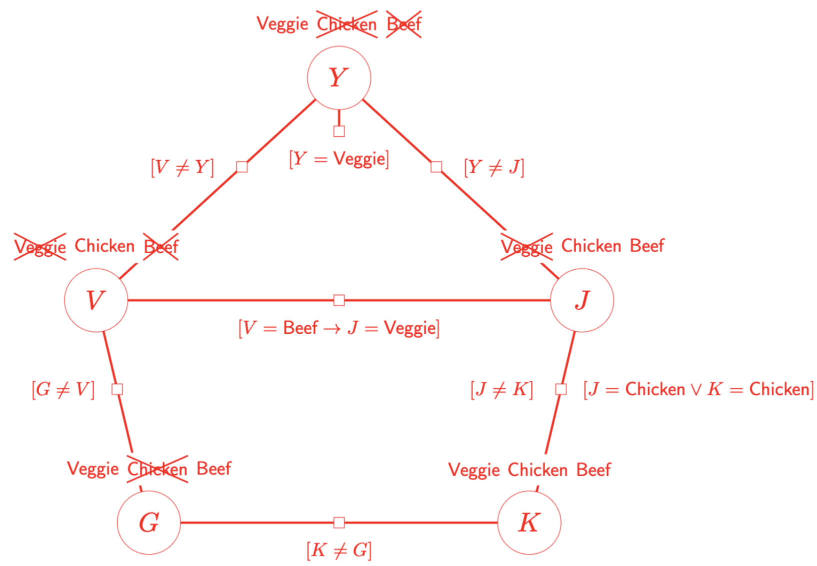
i. Each person must order something different than the people sitting immediately next to them.
ii. You (Y) are vegetarian.
iii. If Veronica (V) orders beef, then Jarvis (J) will order veggie.
iv. Kanti (K) and Jarvis (J) cannot both get non-chicken dishes.

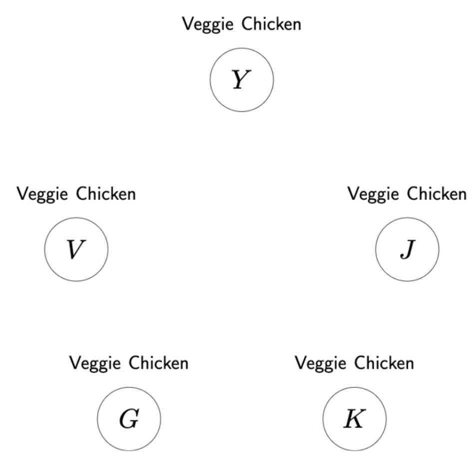
Draw in factors into the above figure based on the above constraints, and write the propositional formula above each factor (e.g., $[Y = \text{Veggie}]$). Then for each pair of variables, enforce arc consistency in both directions, crossing out the appropriate values from the domains.

(b) Your server comes by your table and tells you that they are out of beef today, so you all decide to rework your constraints. Now they are:

i. There is a preference for people sitting next to each other to order different dishes. Formally, we have 5 factors: $f(Y, J) = \mathbf{1}[Y \neq J] + 1$, $f(J, K) = \mathbf{1}[J \neq K] + 1$, etc.
ii. You (Y) are vegetarian.

With the 2 constraints above, what is a maximum weight assignment and what is its weight? If there are many assignments with the same maximum weight, give any one. For convenience, the updated table is given below.





Solution The maximum weight 16. Many answers can be correct. They just need to satisfy the following: 1. Y=Veggie; 2. Only one pair of adjacent people have the same dish. One max weight assignment is: Y=Veggie, V=Chicken, G=Veggie, K=Chicken, and J=Chicken

3) **[CA session] Problem 3**

After finally meeting up, Romeo (R) and Juliet (J) decide to try to catch a goose (G) to keep as a pet. Eventually, they chase it into a 3×3 hedge maze shown below. Now they play the following turn-based game:

(a) The Goose moves either Down or Right.

(b) Romeo moves either Up or Right.

(c) Juliet moves either Left or Down.

G	o	J
	WALL	
R		

Participants: Goose (G), Romeo (R), Juliet (J), bread (o)

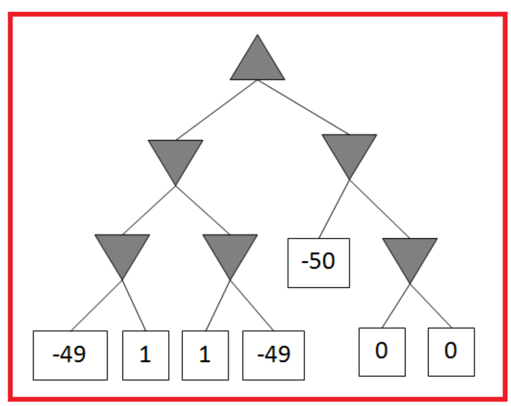
If the Goose enters the square with bread, it gets a reward 1. If either Romeo or Juliet enters the same square as the Goose, they catch it and the Goose gets a reward of -50. The game ends when either the Goose has been caught or everyone has moved once. Note that it is possible for the Goose to get both rewards.

Construct a depth one minimax tree for the above situation, with the Goose as the maximizer and Juliet and Romeo as the minimizers. Use up-triangles Δ for max nodes, down-triangles ∇ for min nodes, and square nodes for the leaves. Label each node with its minimax value.

What is the minimax value of the game if Romeo defects and becomes a maximizer?

Solution

The value of the game is -49 (the goose might as well go for the bread before it gets caught). If Romeo defects, then the value of the game is 0 (the Goose moves towards Romeo).



4) [CA Session] Problem 4

You're programming a self-driving car that can take you from home (position 1) to school (position n). At each time step, the car has a current position $x \in \{1, \ldots, n\}$ and a current velocity $v \in \{0, \ldots, m\}$. The car starts with $v = 0$, and at each time step, the car can either increase the velocity by 1, decrease it by 1, or keep it the same; this new velocity is used to advance x to the new position. The velocity is not allowed to exceed the speed limit m nor return to 0.

In addition, to prevent people from recklessly cruising down Serra Mall, the university has installed speed bumps at a subset of the n locations. The speed bumps are located at $B \subseteq \{1, \ldots, n\}$. The car is not allowed to enter, leave, or pass over a speed bump with velocity more than $k \in \{1, \ldots, m\}$. **Your goal is to arrive at position n with velocity n in the smallest number of time steps.**

Now let's add more information to this problem:
The university wants to remove the old speed bumps and install a single new speed bump at location $b \in \{1, \ldots, n\}$ to maximize the time it takes for the car to go from position 1 to n.

Let $T(\pi, B)$ be the time it takes to get from 1 to n if the car follows policy π if speed bumps B are present. If π violates the speed limit, define $T(\pi, B) = \infty$.

To simplify, assume $n = 6$ and $k = 1$. Again, there is exactly one speed bump. That is, $B = \{b\}$ with $b \in \{1, \ldots, n\}$.

$x = 1$ home	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$ school

Figure: The university will add a speed bump somewhere

(i) [5 points] Compute the worst case driving time, assuming you get to adapt your policy to the university's choice of speed bump location b: $\max_b \min_\pi T(\pi, \{b\})$. What values of b attain the maximum?

Solution Note that with $n = 6$, there are only two places where one can travel at a velocity of 2, from 2 to 4 or 3 to 5; in these cases, there can't be any speed bumps there. So if the speed bump is placed at $b \in \{1, 2, 5, 6\}$, the optimal policy has space to speed up to a velocity of 2 around the bump, so the total time is 4. However, if the speed bump is placed at $\boxed{b \in \{3, 4\}}$, then the optimal policy is to travel at a velocity of 1 the whole way which results in a total time of $\boxed{5}$, which is the worst case. Most common error was missing one of the cases for b. Also, there were a number of off-by-one errors (takes only 5 units to get from 1 to 6, not 6).

(ii) [5 points] Compute the best possible time assuming that you have to choose your policy before the university chooses the speed bump: $\min_\pi \max_b T(\pi, \{b\})$. Make sure to explain your reasoning.

Solution If we choose any policy that has velocity of 2, the university can place the speed bump in the appropriate place that results in a time of ∞. Therefore, we must choose a policy that only has velocity 1, which results in a time of $\boxed{5}$. Students should not assume that the university will definitely place speed bumps at $b \in \{3, 4\}$, but it's fine to acknowledge this as a possibility in your reasoning.