JENNIFER SHI
20662230
j77shi@uwaterloo.ca
CS486 AI

1 Wood Stick Grid Puzzle

a)

	1	2	3	2	
1	4	3	2	1	4
3	2	1	3	4	1
3	1	2	4	3	2
2	3	4	1	2	2
	2	1	2	3	

Strategy 1: **(Viewing constraint)** When I see there is 1 as constraint, (i.e. row=1, col=1) I place 4 to the row/column associated with it, since 4 will block all other sticks.

Strategy 2: **(Viewing constraint)** When I see there is 4 as constraint, (i.e. row=1, col=4). I place the sticks the order [1,2,3,4], since all other orders will not enable the viewer to see exactly 4 different sticks.

Strategy 3: **(All different constraints)** When I see there is a column/row missing 1 number, I found the remaining different sticks to place in. For example, in the figure below, I know in ???, it must be number 2, since this is the only way that will allow column 4 to have sticks that are all different.

	1	2	3	2	
1	x	x	x	1	4
3	x	x	x	4	1
3	x	x	x	3	2
2	x	x	x	???	2
	2	1	2	3	

b) Domains: domain for all of Ri and Cj where i,j = {1,2,3,4} is all the possible combinations of [{1,2,3,4}{1,2,3,4}{1,2,3,4}{1,2,3,4}], There are total of 4*4*4*4 possibilities.

Constraints:
Unary Constraints:
1) All different Constraints, the numbers in each col and row must be all different.
2) Viewing constraints, each row and col have two numbers associated with it. Each number specifies the exact number of different sticks that a viewer can see from the side. For example, the number on the left side of R4 is 2 and the number on the right side of R4 is 2. Then, [1,4,2,3] is a possible placement. And [1,2,3,4] is not, since the person on the left will be able to see 4 sticks, and the person on the right will be able to see only 1 stick.

Binary Constraints:

1) Shared Value Constraint, between each row and column, there is one value that is shared. For example, the coordinate (4,1) is shared between Row4 and Col1, that is, the first value of row4 and the 4^{th} value of col1 is shared. They must be identical. So, if the domain set of row4 is { [1,2,3,4], [1,2,3,4] }, then we can see that the first value of R4 can only be 1. Thus, the 4^{th} value of C1 can also only be 1. If the set of C1 is { [2,3,4,1], [2,4,1,3] }, then we can remove [2,4,1,3] from the domain of C1, since the 4^{th} value is 3 and not 1.

c) After processing unary constraints

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243, 1342, 2341}

R4 ∈ {1423, 2143, 2413, 3142, 3241, 3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {1243, 1342, 2341}

C4 ∈ {1432, 2431, 3421}

JENNIFER SHI j77shi@uwaterloo.ca
20662230

d) Execute the AC-3 algorithm for 8 steps below:

(1) Remove $\langle R_4, (R_4, C_1)\rangle$ from S.
Remove 2413, 1423, 2143 from the domain of R_4.
Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243, 1342, 2341}

R4 ∈ {3142, 3241, 3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {1243, 1342, 2341}

C4 ∈ {1432, 2431, 3421}

(2) Remove $\langle R_4, (R_4, C_2)\rangle$ from S.

Remove 3142, 3241 from the domain of R4.

Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243, 1342, 2341}

R4 ∈ {3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {1243, 1342, 2341}

C4 ∈ {1432, 2431, 3421}

(3) Remove $\langle C_4, (R_1, C_4) \rangle$ from S.

Remove 2431, 3421 from the domain of C4.

Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243, 1342, 2341}

R4 ∈ {3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {1243, 1342, 2341}

C4 ∈ {1432}

(4) Remove $\langle C_3, (R_4, C_3) \rangle$ from S.

Remove 1243, 1342 from domain of C3.

Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243, 1342, 2341}

R4 ∈ {3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {2341}

C4 ∈ {1432}

(5) Remove $\langle R_3, (R_3, C_4)\rangle$ from S.

Remove 1342, 2341 from the domain of R3.

Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {1324, 2134, 2314}

R3 ∈ {1243}

R4 ∈ {3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {2341}

C4 ∈ {1432}

(6) Remove $\langle R_2, (R_2, C_3)\rangle$ from S.

Remove 1324, 2314 from the domain of R2.

Add nothing back to S.

R1 ∈ {4321}

R2 ∈ {2134}

R3 ∈ {1243}

R4 ∈ {3412}

C1 ∈ {4123, 4213}

C2 ∈ {3124, 3214}

C3 ∈ {2341}

C4 ∈ {1432}

(7) Remove ⟨C_1, (R_2, C_1)⟩ from S.

Remove 4123 from the domain of C1.

Add <R4, (R4, C1)> back to S.

R1 ∈ {4321}

R2 ∈ {2134}

R3 ∈ {1243}

R4 ∈ {3412}

C1 ∈ {4213}

C2 ∈ {3124, 3214}

C3 ∈ {2341}

C4 ∈ {1432}

(8) Remove ⟨C_2, (R_2, C_2)⟩ from S

Remove 3214 from the domain of C2.

Add <R4, (R4, C2)> back to S.

R1 ∈ {4321}

R2 ∈ {2134}

R3 ∈ {1243}

R4 ∈ {3412}

C1 ∈ {4213}

C2 ∈ {3124}

C3 ∈ {2341}

C4 ∈ {1432}

After the 8 steps, we need to continue executing the algorithm. We cannot terminate the algorithm because the set S is not yet empty. Although it seems like we have found a solution where the domains of all the variables only have 1 value left, we still need to verify the remaining values do not violate any of the binary constraints by continue executing the algorithm and remove all the arcs from set S.

e) The two approaches are similar as they all use strategies relating to the constraints to find the solution. However, when we solving it by hand, we fill the puzzle out by each number of a specific row/column, not as the entire row or column. They are different in a way that the AC-3 algorithm tends to consider the entire row and column in a “reduction method”, where we reduce the possibilities of the variable and when we have only 1 possible combination left, that is the solution. When we are solving by hand, in the contrast, we are doing a “deterministic method”, where we pick numbers when we are certain that that is the solution, we do not reduce the possible cases, we do not pay attention to all the possible combinations. We focus on the current step and not the puzzle as a whole.

2 Decision Trees (75 marks)

Part (a)
1) tree-full.png is generated after running the program
2) The maximum depth of tree-full is 17
3) The prediction accuracy is 100%.

Part (b)
1) cv-max-depth.png is generated after running the program
2) The best value of the maximum depth is 6 based on the highest average prediction accuracy on the validation set. The average prediction accuracy: 0.5634694333103643
3) tree-max-depth.png is generated with max-depth 6
4) prediction accuracy of the tree-max-depth on the entire data-set: 0.6372732958098811

Part (c)
1) cv-min-info-gain.png is generated after running the program
2) the best value of the minimum information gain pruning criterion is: 1
3) tree-min-info-gain.png is generated after running the program using the entire data-set
4) The prediction accuracy of the decision tree tree-min-info-gain on the entire data-set: 0.7535959974984365

Part (d)

I would generate a decision tree first by using the cross validation to find out the best maximum-depth, and generate a max-depth decision tree. Further, I will prune the tree with min-info-gain for further improvement. We can see that max-depth decision tree provides a reasonable prediction accuracy on the validation set and it is a simple model with the fewer nodes. We should always use cross validation to avoid overfitting.