

## Project 4: Constraint Satisfaction Problems

CSPs are a set of problems in artificial intelligence that are defined as a set of non-atomic units that have states which satisfy a set of constraints. The fact that CSPs consist of non-atomic units is important, as it is what distinguishes them from previously examined problems. More concretely CSPs consist of a set of variables, a set of domains for each of those variables, and a set of constraints that define the scope of the variables and domains.

**For each of the two problems below specify variables, domains, and constraints. Suggest possible methods of a solutions.**

1. **Rectilinear floor-planning:** find non-overlapping places for a number of smaller rectangles
  - a. Say the main rectangle has height  $H$  and width  $W$
  - b. Variables for each rectangle
    - i. Height
    - ii. Width
    - iii. X-position
    - iv. Y-position
  - c. Domains for the variables
    - i. Height: all values  $\geq 0$  and all values  $\leq H$
    - ii. Width: all values  $\geq 0$  and all values  $\leq W$
    - iii. X-position: all values such that the X-position + width is  $\leq W$
    - iv. Y-position: all values such that the Y-position + height is  $\leq H$
  - d. Constraints for the problem
    - i. No rectangles can overlap
    - ii. For any  $i$  and  $j$ ,  $R_i$  and  $R_j$  are not equal, where  $R$  defines the height, width, X-position, and Y-position of a rectangle.
  - e. Possible solution method
    - i. Backtracking could work well pretty well with this problem as it is an algorithm that throws away a “bad path” when configurations are being generated. Although for a large rectangle or a rectangle with a large number of smaller rectangles might take very long to generate. Nonetheless this would work much better than a brute-force approach.
2. **Class scheduling:** there is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes to teach.
  - a. Variables
    - i. classes
    - ii. rooms
    - iii. professors

- b. Domains for each of the variables
  - i. The times that the classes are available
  - ii. The rooms that are available in the university
  - iii. The classes that the professors are able to teach
- c. Constrains for the problem
  - i. A class cannot be taken alongside a prerequisite
  - ii. Professors cannot teach two classes at once
  - iii. Two class cannot be offered in the same room at the same time
- d. Possible solution method
  - i. I could see this being solved with a graph. The graph can be structured in such a way as to represent the possible classes the professor could teach in any given time. The graph must be traversed in such a way as to maximize the classes being taught without overlapping time slots. A minimum spanning tree, perhaps? Solutions to this problem can be generated in a brute force way as well, where all possible class sequences that a professor can teach are generated, and the sequence with no overlapping time slots or rooms is selected.

**Consider “the missionary and cannibal problem”: Three missionaries and three cannibals come to the bank of a river they wish to cross. There is a boat that will hold any two, and any of the group is able to row. If there are more missionaries than cannibals on any side of the river the cannibals will get converted. Deices a series of moves to get all the people across the river with no conversions.**

A, B, C represent the cannibals.

X, Y, Z represent the missionaries.

Travel number	Starting position	Traveling	Ending Position
0	A,B,C ; X,Y,Z		
1	B,C ; Y,Z	A,X =>	
2	B,C ; Y,Z	<= X	A
3	B,C ; Z	X, Y =>	A
4	B,C ; Z	<= Y	A, X
5	C ; Z	B,Y =>	A, X
6	C; Z	<= Y	A, B, X
7	C	Y,Z =>	A, B, X
8	C	<= Z	A, B, X, Y
9		C,Z =>	A, B, X, Y
End			A, B, C ; X, Y, Z