Co-financed by the Connecting Europe Facility of the European Union

Prof. Dr. techn. Wolfgang Nejdl
Franziska Schoger

October 18th 2023

---

# Advanced Topics in AI

## Exercise 2 - Constraint Satisfaction Problems

## Question 1: Course Scheduling

You are in charge of scheduling classes. There are 5 classes given by three teachers and three time slots. You are constrained by the fact that each teacher can only teach one class at a time.

The classes are:

- Class 1 - AI: taught by Prof. A
- Class 2 - Trustworthy AI: taught by Prof. A and Prof. E together
- Class 3 - Explainable AI: taught by Prof. A
- Class 4 - Ethics: taught by Prof. E
- Class 5 - Healthcare Management: taught by Prof. H

The time slots are:

- T1: Saturday 15:00 - 16:30, but Prof. H has another appointment.
- T2: Saturday 16:30 - 18:00, Prof. A can only half of the time ($C_2$ would be possible).
- T3: Wednesday 18:00 - 19:30, but Prof. E has another appointment.

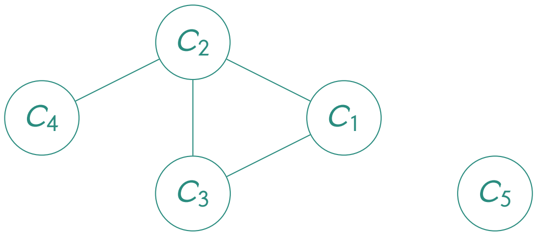
a. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains (after enforcing unary constraints), and binary constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

**Solution:**

| Variables | Domains (or unary constrains) | Binary Constraints |
|---|---|---|
| $C_1$ | $\{T1, T3\}$ | $C_1 \neq C_2$ |
| $C_2$ | $\{T1, T2\}$ | $C_1 \neq C_3$ |
| $C_3$ | $\{T1, T3\}$ | $C_2 \neq C_3$ |
| $C_4$ | $\{T1, T2\}$ | $C_2 \neq C_4$ |
| $C_5$ | $\{T2, T3\}$ | |

b. Draw the constraint graph associated with your CSP.

**Solution:**



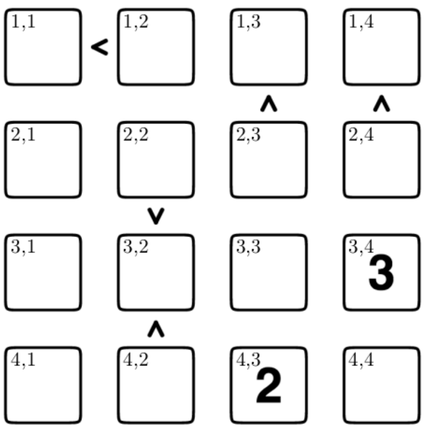

c. State one solution of this CSP.

**Solution:**

This is one solution, other solutions are also possible.

- Class 1 - T1 (can be changed with C3)
- Class 2 - T2
- Class 3 - T3 (can be changed with C1)
- Class 4 - T1
- Class 5 - T2 or T3

## Question 2: Futoshiki




Futoshiki is a Japanese logic puzzle that is very simple, but can be quite challenging.
You are given an $n \times n$ grid, and must place the numbers $1, \ldots n$ in the grid such that every row and column has exactly one of each. Additionally, the assignment must satisfy the inequalities placed between some adjacent squares.

Left is an instance of this problem, for size $n = 4$. Some of the squares have known values, such that the puzzle has a unique solution.

Note also that inequalities apply only to the two adjacent squares, and do not directly constrain other squares in the row or column.

Let's formulate this puzzle as a CSP. We will use $4^2$ variables, one for each cell, with $X_{ij}$ as the variable for the cell in the $i$th row and $j$th column (each cell contains its $i, j$ label in the top left corner). The only unary constraints will be those assigning the known initial values to their respective squares (e.g. $X_{34} = 3$).

a. Complete the formulation of the CSP using only unary and binary constraints. In particular, describe the domains of the variables, and all binary constraints you think are necessary. You do not need to enumerate them all, just describe them using concise mathematical notation. You are not permitted to use $n$-ary constraints where $n \geq 3$

**Solution:**

Domains: $X_{\text{ij}} \in \{1, 2, 3, 4\}, \forall i, j$
Unary constraints: $X_{34} = 3, X_{43} = 2$
Inequality binary constraints: $X_{11} < X_{12}, X_{13} < X_{23}, X_{14} < X_{24}, X_{32} < X_{22}, X_{32} < X_{42}$
Row binary contraints:$X_{\text{ij}} \neq X_{\text{ik}}, \forall i, j, k, j \neq k$
Column binary contraints:$X_{\text{ij}} \neq X_{\text{kj}}, \forall i, j, k, i \neq k$

b. After enforcing unary constraints, consider the binary constraints involving $X_{14}$ and $X_{24}$. Enforce arc consistency on just these constraints and state the resulting domains for the two variables.

**Solution:**

$X_{14} \in \{1, 2\}, X_{24} \in \{2, 4\}$. Note that both threes are removed from the column constraint with $X_{34}$.

c. Suppose we enforced unary constraints and ran arc consistency on this CSP, pruning the domains of all variables as much as possible. After this, what is the maximum possible domain size for any variable? [Hint: consider the least constrained variable(s); you should not have to run every step of arc consistency.]

**Solution:**

The maximum possible domain size is 4 (ie, no values are removed from the original domain). Consider $X_{21}$ – we will not be able to eliminate any values from its domain through arc consistency.

d. Suppose we enforced unary constraints and ran arc consistency on the initial CSP in the figure above. What is the maximum possible domain size for a variable adjacent to an inequality?

**Solution:**

The maximum domain size is 3 – you must always eliminate either 1 or 4 from a variable participating in an inequality constraint.

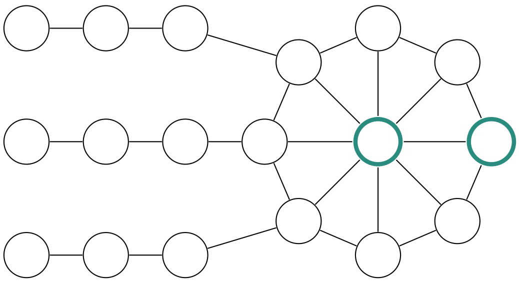
e. By inspection of column 2, we find it is necessary that $X_{32} = 1$, despite not having found an assignment to any of the other cells in that column. Would running arc consistency find this requirement? Explain why or why not.

**Solution:**

No, arc consistency would not find this requirement. Enforcing the $X_{32} \implies X_{42}$ and the $X_{42} \implies X_{43}$ arc leaves $X_{42}$ with a domain of $\{3, 4\}$. Enforcing the $X_{32} < X_{22}$ constraints and $X_{32} \neq X_{34}$ leaves $X_{32} \in \{1, 2\}$ and $X_{22} \in \{2, 3, 4\}$. Enforcing that they are all different does not remove any values. After this point, every arc in this column is consistent and $X_{32}$ is not required to be 1.

## Question 3: Constraint Graph

Consider the following constraint graph:



In two sentences or less, describe a strategy for efficiently solving a CSP with this constraint structure.

**Solution:**

Loop over assignments to the variables shown in colour. Treating these nodes as a cutset, the graph becomes one tree-structured CSP, which can be solved efficiently.

## Question 4: Time Management

Two teaching assistants, Ann and Bob, are making their schedules for a busy morning. There are five tasks to be carried out:

- F Pick up food for the group's research seminar, which, sadly, takes one precious hour.
- H Prepare homework questions, which takes 2 consecutive hours.
- P Prepare a robot for a group of preschoolers' visit, which takes one hour.
- S Lead the research seminar, which takes one hour.
- T Teach the preschoolers about the robot, which takes 2 consecutive hours.

The schedule consists of one-hour slots: 8am-9am, 9am-10am, 10am-11am, 11am-12pm. The requirements for the schedule are as follows:

(a) In any given time slot each person can do at most one task (F, H, P, S, T).

(b) The robot preparation (P) should happen before teaching the preschoolers (T).

(c) The food should be picked up (F) before the seminar (S).

(d) The seminar (S) should be finished by 10am.

(e) Ann is going to deal with food pick up (F) since she has a car.

(f) The person not leading the seminar (S) should still attend, and hence cannot perform another task (F, T, P, H) during the seminar.

(g) The seminar (S) leader does not teach the preschoolers (T).

(h) The person who teaches the preschoolers (T) must also prepare the robot (P).

(i) Preparing homework questions (H) takes 2 consecutive hours, and hence should start at or before 10am.

(j) Teaching the preschoolers (T) takes 2 consecutive hours, and hence should start at or before 10am.

To formalize this problem as a CSP, use the variables F, H, P, S and T. The values they take on indicate the person responsible for it, and the starting time slot during which the task is carried out (for a task that spans 2 hours, the variable represents the starting time, but keep in mind that the person will be occupied for the next hour also - make sure you enforce constraint (a)!). Hence there are eight possible values for each variable, which we will denote by A8, A9, A10, A11, D8, B9, B10, B11, where the letter corresponds to the person and the number corresponds to the time slot. For example, assigning the value of A8 to a variables means that this task is carried about by Ann from 8am to 9am.

**Solution:**

Variables: F, H, P, S, T
Domain(s): $\{XY \mid X \in \{A, B\}, Y \in \{8, 9, 10, 11\}\}$
Constraints:

(a) Alldiff(F, H, P, S, T) (B)

(b) $P_Y < T_Y$ (B)

(c) $F_Y < S_Y$ (B)

(d) $S_Y < 10$ (U)

(e) $F_X = A$ (U)

(f) $S_Y \neq F_Y, H_Y, P_Y, T_Y$ (B)

(g) $S_X \neq T_X$ (B)

(h) $T_X = P_X$ (B)

(i) $H_Y \leq 10$ (U) and

$$H_Y \neq \begin{cases} F_Y, F_Y - 1, & if\ H_X = F_X \\ P_Y, P_Y - 1, & if\ H_X = P_X \\ S_Y, S_Y - 1 & \\ T_Y, T_Y - 1, & if\ H_X = T_X \end{cases} \quad \text{(B)}$$

(j) $T_Y \leq 10$ (U) and

$$T_Y \neq \begin{cases} F_Y, F_Y - 1, & if\ T_X = F_X \\ H_Y, H_Y - 1, & if\ T_X = H_X \\ P_Y, P_Y - 1, & if\ T_X = P_X \\ S_Y, S_Y - 1 & \end{cases} \quad \text{(B)}$$

a. What is the size of the state space for this CSP?

**Solution:**

$8^5$

b. Which of the statements above include unary constraints?

**Solution:**

(d), (e), (i), (j). (i) and (j) are both unary constraints, and binary constraints in a single sentence.

c. In the table below, enforce all unary constraints by crossing out values in the table on the left below.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |
| H | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |
| P | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |
| S | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |
| T | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |

**Solution:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | A8 | A9 | A10 | A11 | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| H | A8 | A9 | A10 | ~~A11~~ | B8 | B9 | B10 | ~~B11~~ |
| P | A8 | A9 | A10 | A11 | B8 | B9 | B10 | B11 |
| S | A8 | A9 | ~~A10~~ | ~~A11~~ | B8 | B9 | ~~B10~~ | ~~B11~~ |
| T | A8 | A9 | A10 | ~~A11~~ | B8 | B9 | B10 | ~~B11~~ |

d. Start from the table above, select the variable S and assign the value A9 to it. Perform forward checking by crossing out values in the table below.

**Solution:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | A8 | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| H | ~~A8~~ | ~~A9~~ | A10 | ~~A11~~ | ~~B8~~ | ~~B9~~ | B10 | ~~B11~~ |
| P | A8 | ~~A9~~ | A10 | A11 | B8 | ~~B9~~ | B10 | B11 |
| S | ~~A8~~ | A9 | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| T | ~~A8~~ | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | B10 | ~~B11~~ |

e. Based on the result of 4), what variable will we choose to assign next based on the MRV heuristic (breaking ties alphabetically)? Assign the first possible value to this variable, and perform forward checking by crossing out values in the table below.
Have we arrived at a dead end (i.e., has any of the domains become empty)?

**Solution:**

Variable F is selected and gets assigned value A8 .

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | A8 | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| H | ~~A8~~ | ~~A9~~ | A10 | ~~A11~~ | ~~B8~~ | ~~B9~~ | B10 | ~~B11~~ |
| P | ~~A8~~ | ~~A9~~ | A10 | A11 | B8 | ~~B9~~ | B10 | B11 |
| S | ~~A8~~ | A9 | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| T | ~~A8~~ | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | B10 | ~~B11~~ |

No dead end.

f. We return to the result from enforcing just the unary constraints, which we did in 3). Select the variable S and assign the value A9. Enforce arc consistency by crossing out values in the table below.

**Solution:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | A8 | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| H | ~~A8~~ | ~~A9~~ | A10 | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| P | ~~A8~~ | ~~A9~~ | ~~A10~~ | ~~A11~~ | B8 | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| S | ~~A8~~ | A9 | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | ~~B10~~ | ~~B11~~ |
| T | ~~A8~~ | ~~A9~~ | ~~A10~~ | ~~A11~~ | ~~B8~~ | ~~B9~~ | B10 | ~~B11~~ |

g. Compare your answers to 4) and to 6). Does arc consistency remove more values or less values than forward checking does? Explain why.

**Solution:**

Arc consistency removes more values. It's because AC checks consistency between any pair of variables, while FC only checks the relationship between pairs of assigned and unassigned variables.

h. Check your answer to 6). Without backtracking, does any solution exist along this path? Provide the solution(s) or state that there is none.

**Solution:**

AC along this path gives 1 solution: F: A8 H: A10 P: B8 S: A9 T: B10

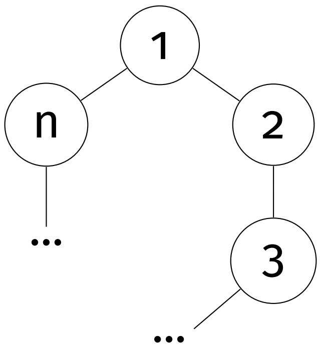
i. Suppose we have just one teaching assistant and $n$ tasks, where no two tasks can be done simultaneously. We are assured that the constraint graph will always be tree-structured and that a solution exists. What is the runtime complexity in terms of the number of tasks, $n$, of a CSP solver that runs arc-consistency and then assigns variables in a topological ordering?

**Solution:**

$O(n^3)$. Modified AC-3 for tree-structured CSPs runs arc-consistency backwards and then assigns variables in forward topological (linearized) ordering so that we don't have to backtrack. The runtime complexity of modified AC-3 for tree-structured CSPs is $O(nd^2)$, but note that the domain of each variable must have a domain of size at least $n$ since a solution exists.

## Question 5: Circular Structured CSPs

Consider a CSP forming a circular structure that has $n$ variables $(n > 2)$, as shown below. Also assume that the domain of each variable has cardinality $d$ ($d$ entries).




a. Explain precisely how to solve this general class of circle-structured CSPs efficiently (i.e. in time linear in the number of variables), using methods covered in class. Your answer should be at most two sentences.

**Solution:**

We fix $X_j$ for some $j$ and assign it a value from its domain (i.e. use cutset conditioning on one variable). The rest of the CSP now forms a tree structure, which can be efficiently solved without backtracking by enforcing arc consistency. We try all possible values for our selected variable $X_j$ until we find a solution.

b. If standard backtracking search were run on a circle-structured graph, enforcing arc consistency at every step, what, if anything, can be said about the worst-case backtracking behavior (e.g. number of times the search could backtrack)?

**Solution:**

A tree structured CSP can be solved without any backtracking. Thus, the above circle-structured CSP can be solved after backtracking at most $d$ times, since we might have to try up to d values for $X_j$ before finding a solution.