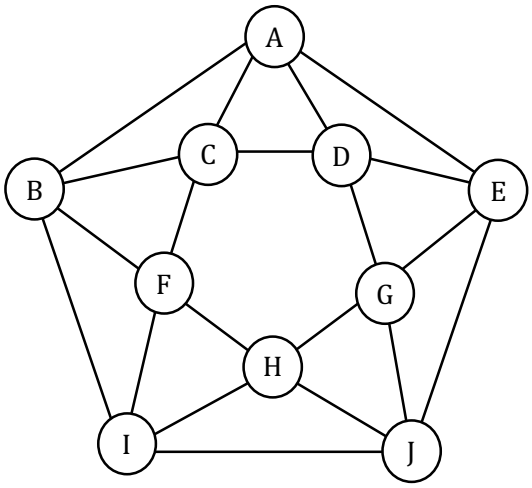
Constraint Programming

2017/2018– Mini-Test #1
Thursday, 2 November, 11:00 h in Room 114-II
Duration: 1.5 h (open book)

1. Finite domain Constraints - Propagation (7 pts)



Consider the constraint network on the right, where nodes represent variables, all with domain {**1,2,3}**. Arcs represent different constraints (≠).

a) **(2 pt)** Is the problem satisfiable? Justify your answer.

b) **(1 pt)** What pruning is achieved initially, if node-consistency is maintained? And arc-consistency?

c) **(2 pt)** What pruning is achieved initially, if path-consistency is maintained?

d) **(2 pt)** Assume now a problem with the same structure (variables and constraints) but where the variables and the constraints are arbitrary. Show that the problem is satisfiable if it is path consistent after labelling variables A and C. Justify your answer.

2. Modelling with Finite Domain Constraints (8 pts)

Ian lives with his wife Lou, their two sons, Joe and Ken, and their mothers (Ian's mother, Mia, and Lou's mother's, Sue). He bought a set of ten birthday candles, each with a distinct digit. Then he noticed the coincidence that he could use each candle exactly once to celebrate the 6 birthdays of the family in this year. What are the ages of the six members of the family (after their birthday), given that a) no one had children before 19 nor after 32, b) the age difference between Ian and his wife Lou is no more than 2 years, and c) the age difference between the brothers is only one year.

a) **(4 pt)** Specify a model for this problem in Comet, namely declaring the decision variables you use in the model (with their domains), as well as the constraints that should be posted to impose the restrictions of the problem.

b) **(2 pt)** For the model you adopted, are there symmetric solutions, i.e. solutions that can be obtained from others by a simple mapping? If so, can you add extra constraints that prevent searching for these symmetric solutions.

c) **(2 pt)** Adapt the above problem, for another family with the same structure (for simplicity we keep the same names) for any age difference (≥ 1) between the brothers and where the 10 candles can be used exactly twice over two consecutive years.

3. Global Constraints (5 pts)

Consider a function that constrains an array of `n` decision variables, `a`, that take values in the range `1..m`, such that the maximum difference between the number of occurrences in a solution of any values in this range is less or equal to `k`. For example, array `a = [2,2,1,1,2,2]` does not satisfy the constraint when `m = 3` and `k = 2` since the difference between the number of occurrences of the values 2 (that occurs 4 times) and 3 (0 times) is **4**, hence greater than `k = 2`. However, it satisfies the constraint for `m = 2` and `k = 2`.

a) **(2 pts)** Implement in Comet this "global constraint" with the signature below

```
function void balanced(var<CP>{int} [] a, int m, int k) {…}
```

b) **(2 pts)** If your implementation of this global constraint would maintain domain consistency, i.e. it pruned any values from the domain of the variables that do not occur in any solution, what would be such pruning for `n = 8`, `m = 5`, `k = 2` and constraint variables with domains

```
a[1] in 1..4;     a[2] in 2..3;     a[3] in 2..3;     a[4] in 1..4;
a[5] in 1..4;     a[6] in 2..3;     a[7] in 2..3;     a[8] in 1..4;
```

c) **(1 pts)** Assume that the constraint also imposes the array a to be monotonically increasing (i.e. `forall(i in 1..n-1) a[i] <= a[i+1]`). Again, assuming that the implementation of the constraint guarantees domain consistency, would there be any further pruning of the domains?