Constraint Solving - Global Constraints

1. Traveling Salesperson (TSP)

The distances between a set cities in Bavaria are specified in files **bavariaNN.txt** (where NN represents the number of cities considered), in **Bavaria_benchmarks.zip**[†].

The files start by the number **k** of cities, followed by the adjacency matrix that constrains the distances between all pairs of cities. For example, the file "bavaria07.txt" contains the text on the right.

```
7
  0 107 241 190 124  80 316
107   0 148 137  88 127 336
241 148   0 374 171 259 509
190 137 374   0 202 234 222
124  88 171 202   0  61 392
 80 127 259 234  61   0 386
316 336 509 222 392 386   0
```

The TSP problem consists of finding the shortest tour required for a salesman to visit all cities, without visiting any city twice, and returning to the starting city. More formally, considering the graph G = (N,E) where N is the set of k nodes (corresponding to the cities) and E the set of edges between the nodes labelled with their costs (distances in this case), the TSP problem consists of finding the Hamiltonian cycle in the graph G with lowest cost.

Rank: Model (and solve) the problem with array `rank[0..k-1]` of decision variables, where `rank[i]` represents the ith city to be visited in the tour. For example,

`rank = [0,4,1,5,6,3,2]` represents the tour 0→4→1→5→6→3→2→0

Next: Solve the problem with an alternative model using an array `next[1..k]` of decision variables, where `next[i]` represents the city that follows city `i` in the tour. The above solution is now represented by `next = [4,5,0,2,1,6,3].`

In both the above models adopt the symmetry breaking assumption that the tour starts in city 1, and make sure that your solution is not composed of sub-cycles. Which of the models is more efficient?

Global: Solve the TSP problem with the model next, but now using the global constraints **circuit** and **element** available in **Choco**. Compare the efficiency of the execution for various graphs available in file "**bavaria.zip**".

Sugestion for Reading Data Files:

To read a data file with integers with the format above, use the class **graph** that is available in the web page.

To read the adjacency matrix, use method:

- **ReadMatrixFormat(String path)**

The adjacency graph of the graph (a square matrix) is available in the internal variable

- **matriz.**

Its size is available in the internal variable

- **nn.**

[†] **`Source: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`**
`benchmark: bayg29.tsp.gz`

2. Job Shop

A job shop problem consists of scheduling J jobs, each consisting of T tasks, which have precedence constraints. The jobs are independent, except for the fact that the tasks are executed in machines of certain types and there are only a limited number of machines of each type. The goal is to finish all tasks within a certain makespan (Satisfaction) or to minimize the makespan.

i. Solve the job-shop problem for (small) instances obtained from the OR-library[1]:

For example, benchmark " la03.txt", (other benchmarks available in jobshop_benchmarks.zip, together with file read_jshp_mat.co with a function to read this type of files) with the following data:

```
instance la03
Lawrence 10x5 instance (Table 3, instance 3); also called (setf3) or (F3)
10 5
1 23 2 45 0 82 4 84 3 38
2 21 1 29 0 18 4 41 3 50
2 38 3 54 4 16 0 52 1 52
4 37 0 54 2 74 1 62 3 57
4 57 0 81 1 61 3 68 2 30
4 81 0 79 1 89 2 89 3 11
3 33 2 20 0 91 4 20 1 66
4 24 1 84 0 32 2 55 3  8
4 56 0  7 3 54 2 64 1 39
4 40 1 83 0 19 2  8 3  7
```

specifies a problem with 10 jobs (rows) and 5 tasks each, where each row indicates for that job the types of the machines in which the tasks are executed and their duration. For example job 1 is composed of 5 tasks, to be executed, respectively, in machines of type 1,2,0,4 and 3, with duration 23, 45, 82, 84 and 38.

a) Consider the problem variants of satisfaction (finish all tasks before some time T) or minimisation (minimise this time T).

Note: To read the job shop specification adapt the class graph discussed aboce

[1] ORlibrary URL: http://people.brunel.ac.uk/~mastjjb/jeb/info.html. Job shop benchmarks from library (available in the course page) obtained from http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt