Constraint Solving - Global Constraints (1)

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), zipped in **bavaria.zip**[†].

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

```
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[1..k]` of decision variables, where `rank[i]` represents the i-th city to be visited in the tour. For example, tour 1→ 5 → 2 → 6 → 7 → 4 → 3 → 1 is represented by `rank = [1,5,2,6,7,4,3]`.

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 = [5,6,1,3,2,7,4].`

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 **circuitMin** available in Comet. Compare the efficiency of the execution for various graphs available in file "**bavaria.zip**".

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

Sugestion for Reading Data Files:

To read a data file with integers adapt the following procedure that reads a file with data in the format of the data file " **bavariaNN.txt** ", with name **fname**, placed in the same directory of the code file.

```
function int [,] read_mat(Integer N, string fname){
   string [] dirs = System.getArgs();
   string directory = dirs[3].suffix(2);
   string ename = directory + "/" + fname;
   ifstream file(ename);
   N := file.getInt();      // the number of cities
   int d [1..N, 1..N];
   forall(i in 1..N, j in 1 .. N) d[i,j] = file.getInt();
   return d;
}
```