TSPLIB in XML format

To facilitate the usage of the classical TSPLIB by Gerhard Reinelt, we now offer the data of the well-known test instances also in an *XML* data format, both for the symmetric and the asymmetric travelling salesman problem. Thus, the different data representations of the current instances are transformed into a uniform style. This should be helpful also for using the data as a basis for generating instances with additional constraints or other properties. All files have the suffix *.xml*

The data format should be easy to use together with the *XML parsers* provided for the most common programming languages. An example for using a parser is given in the included program contained in the file "Sample.zip". A validation file "TSPConfiguration.xsd" is also provided.

Data format For illustration consider the undirected graph in Figure 1.

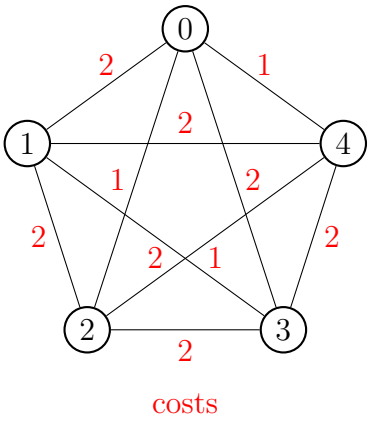

Figure 1: *An instance of the Symmetric travelling salesman problem.*

Listing 1 shows the corresponding XML-file. Lines No. 1 and 2 define the parameters of the *XML* file – e.g. the *version* and the *encoding*. The following tag (line No. 3) specifies the *problem type* and has to be *"travellingSalesmanProblemInstance"*.

The next entries define the *name*, *source* and the *description* of the *TSP* instance. The tag *doublePrecision* states the *number of decimal digits after the comma* of the edge *costs* which are read from the file. The tag *ignoredDigits* represents the *number of remaining decimal digits which have to be ignored when comparing two costs*, in the sense that $10^{-(doublePrecision-ignoredDigits)}$

yields 0 in the computation. Note that all these tags have to be defined and that their relative order is not allowed to vary.

Finally, the weighted graph is defined by the tag *graph* (line No. 9). Each vertex (tags *vertex*) contains the list of all emanating edges. These edges are defined by the tags *edge* (see e.g. the line No. 11). Each edge has an attribute *cost* (always written as a real number) and contains the identifier of the other vertex incident to that edge. The vertex identifiers are defined by their relative order (starting from 0). Note that the cardinality of the vertex set is defined implicitly by the number of *vertex* tags contained in the file. If the graph is undirected, all edges are defined twice (from i to j and from j to i) having the same *costs*.

```
1 <?xml version="1.0" encoding="UTF-8"
2  standalone="ignoredDigitsno" ?>
3 <travellingSalesmanProblemInstance>
4   <name>Small 1</name>
5   <source>Rostislav Stanek</source>
6   <description>A small instance.</description>
7   <doublePrecision>15</doublePrecision>
8   <ignoredDigits>5</ignoredDigits>
9   <graph>
10    <vertex>
11      <edge cost="2.000000000000000e+00">1</edge>
12      <edge cost="1.000000000000000e+00">2</edge>
13      <edge cost="2.000000000000000e+00">3</edge>
14      <edge cost="1.000000000000000e+00">4</edge>
15    </vertex>
16    <vertex>
17      <edge cost="2.000000000000000e+00">0</edge>
18      <edge cost="2.000000000000000e+00">2</edge>
19      <edge cost="2.000000000000000e+00">3</edge>
20      <edge cost="2.000000000000000e+00">4</edge>
21    </vertex>
22    <vertex>
23      <edge cost="1.000000000000000e+00">0</edge>
24      <edge cost="2.000000000000000e+00">1</edge>
25      <edge cost="2.000000000000000e+00">3</edge>
26      <edge cost="1.000000000000000e+00">4</edge>
27    </vertex>
28    <vertex>
29      <edge cost="2.000000000000000e+00">0</edge>
```

```
30      <edge cost="2.000000000000000e+00">1</edge>
31      <edge cost="2.000000000000000e+00">2</edge>
32      <edge cost="2.000000000000000e+00">4</edge>
33    </vertex>
34    <vertex>
35      <edge cost="1.000000000000000e+00">0</edge>
36      <edge cost="2.000000000000000e+00">1</edge>
37      <edge cost="1.000000000000000e+00">2</edge>
38      <edge cost="2.000000000000000e+00">3</edge>
39    </vertex>
40  </graph>
41</travellingSalesmanProblemInstance>
```

Listing 1: *An instance of the Symmetric travelling salesman problem.*

All costs are written as *real numbers* (also if they were given as integers) and the costs were not rounded. For comparing results with the classical *TSPLIB* data, costs should be rounded first (following the *TSPLIB* rules).

Only for the 12 cases of geographical distances (`ali535`, `bayg29`, `bays29`, `burma14`, `gr137`, `gr202`, `gr229`, `gr431`, `gr666`, `gr96`, `ulysses16`, `ulysses22`) the costs were already rounded (but are still given as real numbers) according to the sophisticated rounding rules required by the *TSPLIB* for coordinates on the globe. Thereby, the instances guarantee the same optimal solution values as listed in the classical *TSPLIB*.

Extensions The data format can be easily extended for many problems related to the *TSP* – e.g. for the *Hamiltonian cycle problem (HCP)*, the *Capacitated vehicle routing problem (CVRP)* or for the *Orienteering problem (OP)*. This can be done e.g. by introducing new general tags (as for the *knapsack constraint* in the *OP*) or by defining new vertex or edge attributes (e.g. *edge capacities*, *vertex profits* etc.). The format can also be used for *non-complete graphs* or extended for *multigraphs*.

January, 2013

Ulrich Pferschy
Rostislav Staněk
Department of Statistics and Operations Research
University of Graz
Universitätsstraße 15
8010 Graz, Austria.
{pferschy, rostislav.stanek}@uni-graz.at