First & last name	Cyril Bousmar
NOMA UCLouvain	634018000

LINFO2266: Advanced Algorithms for Optimization

Project 1: Dynamic Programming

Exercise 1 Modeling the Traveling Salesman Problem

The Traveling Salesman Problem is a famous combinatorial optimization problem where a salesman has to find the shortest tour visiting all cities from a given set $N = \{1, \ldots, n\}$. The salesman can travel between every pair of cities $i, j \in N$, which are separated by a distance d_{ij}. Find a dynamic programming model for the Traveling Salesman Problem.

1. What information is contained in the states of the model?

   An integer representing the city, a double which is the distance between the state and its predecessor, and a list containing all the cities that have yet to be visited.

2. Give the Bellman recurrence equation that computes the optimal value for each state of the model. You can assume that the salesman starts at city 1.

   $$C(i, S) = \min_{j \in S, j \neq i} [C(j, S \setminus \{i\}) + d(j, i)]$$

3. Give the base case(s) of the recurrence, and the case that gives the optimal solution of the problem.

   When no more cities have to be visited from the current state, then it is a base case.
   The optimal solution is given by the case costing less.

4. What is the space and time complexity needed by your dynamic proram algorithm for the TSP (wrt the number of cities n). Justify?

   Time complexity of : $O(n^2 \cdot 2^n)$. There are 2^n subsets and each transitions and computations are $O(n)$.

   Space complexity is $O(n \cdot 2^n)$, since we have a table for each subset.

Exercise 2 Experiments

1. Create a table with the rune time for all the instances of size 8, 10 and 18 in the data/TSP directory. This table must be generated by a new class with a main method in your repository. Specify the characteristics the machine used for the experiment.

TSP Instance Size 8

Instance	Runtime (ms)
0	1
1	1
2	0
3	0
4	1
5	1
6	1
7	1
8	3
9	1

TSP Instance Size 10

Instance	Runtime (ms)
0	3
1	3
2	3
3	3
4	3
5	4
6	3
7	3
8	3
9	3

TSP Instance Size 18

Instance	Runtime (ms)
0	1905
1	1858
2	2267
3	1920
4	1963
5	1903
6	1904
7	1940
8	1819
9	1848

Operating System: Mac OS X
CPU Architecture: aarch64
JVM Version: 21
Available Processors: 10

2. Create a graphic with the number of instances solved over time, for instances of size 8, 10 and 18. The x axis is the time, the y axis are the number of instances. (you may use log-scale axis for clarity). The graphic can be created from the content of the data in the table.