Methods and Models for Combinatorial Optimization

Lab exercise - Part I

L. Brentegani, L. De Giovanni, M. Di Summa

Consider the following combinatorial optimization problem and the proposed Integer Linear Programming (ILP) model. **The exercise is to implement the model using the Cplex API and to test it**. Test should determine the ability of the model to provide exact solutions for the proposed problem: in particular we want to know up to which size (namely the number of holes) the problem can be solved in different amounts of time (up to 0.1 seconds, up to 1 second, up to 10 seconds ...).

1 Problem description

A company produces boards with holes used to build electric frames. Boards are positioned over a machines and a drill moves over the board, stops at the desired positions and makes the holes. Once a board is drilled, a new board is positioned and the process is iterated many times. Given the position of the holes on the board, the company asks us to determine the hole sequence that minimizes the total drilling time, taking into account that the time needed for making an hole is the same and constant for all the holes.

2 The model to be implemented

We can represent the problem on a weighted complete graph $G = (N, A)$, where N is the set of nodes and corresponds to the set of the positions where holes have to be made, and A is the set of the arcs (i, j), $\forall\ i, j \in N$, corresponding to the trajectory of the drill moving from hole i to hole j. A weight c_{ij} can be associated to each arc $(i, j) \in A$, corresponding to time needed to move from i to j. In this graph model, the problem can be seen as finding the path of minimum weight that visits all the nodes. Indeed, since the drill has to come back to the initial position in order to start with the next board, the path has to be a cycle. The problem can be thus seen as determining the minimum weight hamiltonian cycle on G, that is a Travelling Salesman Problem (TSP) on G.

2.1 A network flow formulation

The TSP can be formulated (among others) as a network flow model on G. Arbitrarily select a node in N (call it node 0) as starting node, and let $|N|$ be the amount of its outcoming flow. The idea is to push this amount of flow towards the remaining nodes in such a way that (i) each node (different from 0) receives 1 unit of flow, (ii) each node is visited once, and (iii) the sum of c_{ij} over all the arcs shipping some flow is minimum.

2.2 ILP model

SETS:

- N = graph nodes, representing the holes;
- A = arcs (i, j), $\forall\, i, j \in N$, representing the trajectory covered by the drill to move from hole i to hole j.

PARAMETERS:

- c_{ij} = time taken by the drill to move from i to j, $\forall\, (i, j) \in A$;
- 0 = arbitrarily selected starting node, $0 \in N$.

DECISION VARIABLES:

- x_{ij}= amount of the flow shipped from i to j, $\forall\, (i, j) \in A$;
- y_{ij}= 1 if arc (i, j) ships some flow, 0 otherwise, $\forall\, (i, j) \in A$.

MODEL:

$$
\begin{array}{llll}
\min & \displaystyle\sum_{i,j:(i,j)\in A} c_{ij} y_{ij} & & \\
s.t. & \displaystyle\sum_{j:(0,j)\in A} x_{0j} & = & |N| & \\
& \displaystyle\sum_{i:(i,k)\in A} x_{ik} - \sum_{j:(k,j)\in A} x_{kj} & = & 1 & \forall\ k \in N \setminus \{0\} \\
& \displaystyle\sum_{j:(i,j)\in A} y_{ij} & = & 1 & \forall\ i \in N \\
& \displaystyle\sum_{i:(i,j)\in A} y_{ij} & = & 1 & \forall\ j \in N \\
& x_{ij} & \leq & |N|\ y_{ij} & \forall\ (i,j) \in A \\
& x_{ij} \in \mathbb{Z}_+ & & & \forall\ (i,j) \in A \\
& y_{ij} \in \{0,1\} & & & \forall\ (i,j) \in A
\end{array}
$$