Woodler Formulation - Single Truck

The statement of the use case is on Mip Wise's website: mipwise.com/use-cases/freshly.

In this Notebook, we formulate a simplified version of the Woodler problem in which there is only one truck (that is assumed to have enough capacity to deliver all orders).

Input Data Model

Set of indices

- I set of sites.

Parameters

- cv variable cost of the truck.
- td_{ij} transit distance from site i to j.

Decision Variables

- x_{ij} equal 1 if the truck goes from site i to site j.

Constraints

- Flow balance:

$$\sum_{i} x_{ih} = \sum_{j} x_{hj}, \quad \forall h \in I.$$

  This constraint says that the number of trucks that arrives at Site h must equal the number of trucks that depart from Site h.
- Must deliver every order:

$$\sum_{i} x_{ij} = 1, \quad \forall j.$$

  This constraint says that the number of trucks that arrives at Site j is exactly one.

Objective

The objective is to minimize the total cost.

$$\sum_{ij} cv \cdot td_{ij} \cdot x_{ij}.$$

Final formulation

$$
\begin{array}{ll}
\min & \sum_{ij} cv \cdot td_{ij} \cdot x_{ij} \\
\text{s.t.} & \sum_i x_{ih} = \sum_j x_{hj}, \quad \forall h \in I, \\
& \sum_i x_{ij} = 1, \quad \forall j, \\
& x_{ij} \in \{0,1\}, \quad \forall i,j.
\end{array} \tag{1}
$$

Sub-tour Elimination

Depending on the data set, solutions from the formulation above migh come with sub-tours. This is specially true when there is an arc between any two nodes.

Here is what a solution with sub-tours looks like: $(0 \rightarrow 4 \rightarrow 0)$, $(1 \rightarrow 2 \rightarrow 3 \rightarrow 1)$.

There are two classic ways to prevent sob-tours.

MTZ Formulation

One way to prevent sub-tours is to use the so-called *Miller–Tucker–Zemlin* (MTZ) formulation which consists of adding the following set of constraints to the model above:

$$u_i - u_j + 1 \leq N(1 - x_{ij})$$

Here, N is the number of sites and u_i is an auxilary integer decision variable that specify the position of Site i in the sequence of stopts. For example, if the solution is $0 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 0$, then $u_0 = 1, u_1 = 2, u_3 = 3, u_4 = 4, u_2 = 5$.

The problem if this formulation is that it typically doesn't perform well in practice.

DFJ Formulation

An alternative way to prevent sub-tours is to use the so-called *Dantzig–Fulkerson–Johnson* (DFJ) formulation which consists of adding one constraint for each potential sub-tour.

For example, one way to avoid the sub-tour $0 \rightarrow 4 \rightarrow 0$ of the example above, we can add the following constraint to the model:

$$x_{04} + x_{40} \leq 1.$$

This constraint says that the truck can go from Site 0 to Site 4 or from Site 4 to Site 0, but not both.

Similarly, to avoid the sub-tour $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ we can add the following constraint to the model:

$$x_{12} + x_{23} + x_{31} \leq 2.$$

This constraint says that at most two out of the three arcs $(1,2), (2,3), (3,1)$, can be used.

A generalization of the DFJ constraints goes as follows. Given a strict subset of sites S, we add the following constraint to the model:

$$\sum_{(i,j) \in S} x_{ij} \leq |S| - 1,$$

where $|S|$ represents the number of elements of the set S.

The problem with this approach, as you might guess, is that the number of subsets S can be huge depending on the size of the instance. That's why in practice people typically add these constraints on the fly as needed (using callbacks to be more efficient).