EXACT TSP SOLVER

MIP Model

Adam DeJans
25 Aug. 2021

DECISION VARIABLES

We simply need to decide the path to take, which means the sequence of points to follow. This is identical to a shortest path problem; therefore, assuming that P is the set of points, we define

$$x_{i,j} \in \{0,1\} \; \forall i \in P, \forall j \in P,$$

where $x_{i,j}$ with a value of one will indicate that we need to connect points i and j.

OBJECTIVE

The objective function is identical to shortest path.

Assuming that the distance matrix is D, we get

$$\min \sum_i \sum_j D_{i,j} x_{i,j}$$

CONSTRAINTS

We must ensure a tour, a single closed path covering every vertex exactly once.

For each vertex, then, we must choose precisely one arc going in and one arc going out as in

$$\sum_{j \in P \setminus \{i\}} x_{i,j} = 1 \; \forall i \in P$$

and

$$\sum_{j \in P \setminus \{i\}} x_{j,i} = 1 \; \forall i \in P$$

As it turns out, these two constraints are surprisingly not enough.

DIFFICULTY - SUBTOURS

While the previous constraints guarantee that every vertex is on a tour, there may be more than one such tour. Satisfying the previous constraints, we could get a path 0, 1, 3, 4, 0 and another looping around the rest of the vertices.

These problematic paths are known as subtours and must be eliminated. The key to the elimination is to realize that for any strict subset of nodes, the number of chosen arcs must be less than the number of nodes. For instance, to eliminate the subtour 0, 1, 3, 4, 0, we could add the following constraint:

$$x_{0,1} + x_{1,0} + x_{1,3} + x_{3,1} + x_{3,4} + x_{4,3} + x_{4,0} + x_{0,4} \leq 3$$

With the addition of this constraint, solvers will never include more than three arcs between the four problematic vertices, preventing a subtour among them. In other words, it forces a path entering a cluster of nodes to exit the cluster.

OVERCOMING SUBTOUR DIFFICULTIES

So should we just add all subtour constraints? Programmatically it's very feasible.

The issue is that the resulting model would be unwieldly and many solvers would slow down unacceptably. The trick is to improve the model iteratively. We will use the result of a solver run to choose the constraints to add to the next run.

Birds-eye view:

1. Execute the model with no subtour elimination constraints.
2. If the solver returns a single tour, we are done.
3. If it returns a set of subtours, we add the subtour elimination constraints for each of them, and only them.
4. Eventually all *relevant* subtours are eliminated and the solver returns a tour of the whole graph.

SUCCESSIVE (PARTIAL) SOLUTIONS OF TSP

Iteration	Tour(s)
0	[0, 2]; [1, 7, 8]; [3, 6, 9]; [4, 5]
1	[0, 2, 7]; [1, 8]; [3, 6]; [4, 9, 5]
2	[0, 2, 3, 6, 9, 5, 4, 1, 8, 7]