Gurobi Seminar 5
Traveling Salesman Problem

Demetrios Papazaharias

UB INFORMS Student Chapter
Department of Industrial & Systems Engineering
University at Buffalo, SUNY

November 22, 2019

Welcome!

- Welcome to the fifth installment of our Gurobi series!
- In this workshop we will cover the traveling salesman problem
- At the end of this workshop you will know how to implement lazy cuts in Gurobi
- Materials for this workshop can be found here:

  `github.com/Dpapazaharias1/UB-INFORMS-Gurobi-Seminar`

Contents

User Cuts vs. Lazy Cuts

- In the last session we implemented user cuts for vertex packing
- Recall that user cuts are ordinary constraints which are added to improve the LP relaxation, but do not eliminate feasibly solution. A formulation is valid with our without these cuts.
- **Lazy cuts**, in contrast, represent one portion of the constraint set that is essential in defining the formulation. Lazy cuts are only added once an integer feasible solution is identified.
- Lazy cuts are part of a very large constraint set where most constraints are unlikely to be violated. The term "lazy" is reference to only adding the constraint once it is needed.

Contents

Problem Description

- **Input**: A list of cities and the distances between each pair of cities
- **Output**: Minimum cost route which visits each city and return to the origin
- We define this problem over a digraph $G = (N, A)$ where the nodes represent cities and the distance are represented by arc weights c_{ij}
- There are many versions of the TSP but for this workshop we will focus on the symmetric TSP ($c_{ij} = c_{ji}$)

Building the model

- Define a binary variable for each arc

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in the tour} \\ 0 & \text{otherwise} \end{cases}$$

- Minimize the cost of the tour

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij}$$

- Each city must be visited once

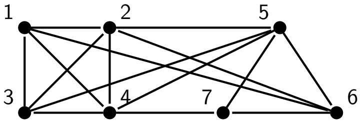
$$\sum_{j\in N} x_{ij} = 1 \quad i \in N$$

- Each city must be departed from once

$$\sum_{i\in N} x_{ij} = 1 \quad j \in N$$

Issues

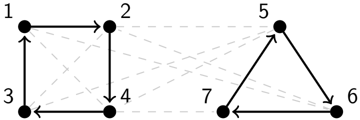
- Consider this instance of the TSP
- If we solve the model as is we may receive the solution:



- We visit and leave each city, but we have **subtours**
- We introduce a set of constraints to break subtours

Issues

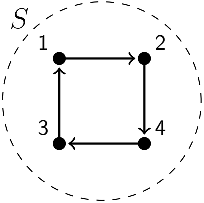
- Consider this instance of the TSP
- If we solve the model as is we may receive the solution:



- We visit and leave each city, but we have **subtours**
- We introduce a set of constraints to break subtours

Subtour Elimination

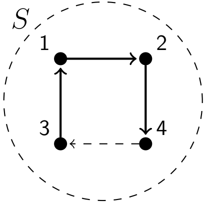
- Consider a subset of vertices S which are part of a subtour



- For any subtour the number of arcs (i, j) with $i, j \in S$ is equal to $|S|$
- We can break these subtours by restricting the number of arcs in S to be $|S| - 1$
- By breaking these subtours we will force one arc to enter S from $V \setminus S$ and one arc from S to $V \setminus S$.

Subtour Elimination

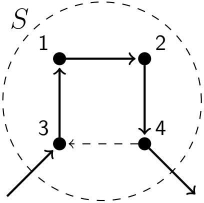
- Consider a subset of vertices S which are part of a subtour



- For any subtour the number of arcs (i, j) with $i, j \in S$ is equal to $|S|$
- We can break these subtours by restricting the number of arcs in S to be $|S| - 1$
- By breaking these subtours we will force one arc to enter S from $V \setminus S$ and one arc from S to $V \setminus S$.

Subtour Elimination

- Consider a subset of vertices S which are part of a subtour



- For any subtour the number of arcs (i, j) with $i, j \in S$ is equal to $|S|$
- We can break these subtours by restricting the number of arcs in S to be $|S| - 1$
- By breaking these subtours we will force one arc to enter S from $V \setminus S$ and one arc from S to $V \setminus S$.

DFJ Formulation

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in the tour} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j\in N} x_{ij} = 1 \quad i \in N \tag{2}$$

$$\sum_{i\in N} x_{ij} = 1 \quad j \in N \tag{3}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - 1 \quad S \subset N \tag{4}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A \tag{5}$$

Subtour Elimination

The complicating constraints for the TSP are the subtour elimination constraints:

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

- There are $O(2^n)$ subtour elimination constraints
- In practice, only a fraction of the subtour elimination constraints are necessary
- In order to solve the TSP, we will begin with none of the constraints and add them as needed

Separation Algorithm

For TSP, we will check if a current integer solution violates any of the subtour elimination constraints:

- Given a solution $\bar{x}_{ij}$, we construct a digraph $G' = (N, A')$ where $A' = \{(i, j) \in A | \bar{x}_{ij} = 1\}$
- Find the connected components of G'
- If there is more than one connected component, then G' contains subtours, add the corresponding constraint for each subtour S:

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

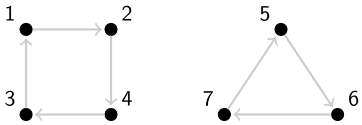
and solve the LP relaxation.

Connected Components

procedure ConnectedComponents
    $C \leftarrow \emptyset$
    initialize $i \in N$ as undiscovered
    for $i \in N$ **do**
        $K, Q \leftarrow \emptyset$
        if i is not discovered **then**
            label i as discovered
            $K \leftarrow K \cup \{i\}$, $Q \leftarrow Q \cup \{i\}$
            while Q is not empty **do**
                $v \leftarrow Q$.dequeue()
                for $u \in$ Adj(v) **do**
                    if u is not discovered **then**
                        label u as discovered
                        $K \leftarrow K \cup \{i\}$, $Q \leftarrow Q \cup \{i\}$
            $C \leftarrow C \cup \{K\}$

Detecting Subtours

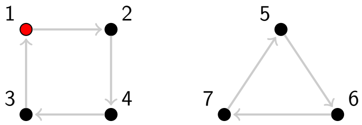
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{0, 0, 0, 0, 0, 0, 0\}$$

Detecting Subtours

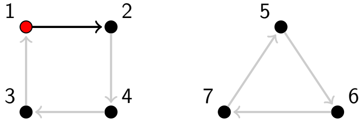
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1\}$$
$$Q = \{1\}$$
$$D = \{1, 0, 0, 0, 0, 0, 0\}$$

Detecting Subtours

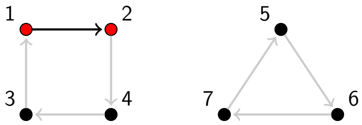
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1\}$$
$$Q = \{\}$$
$$D = \{1, 0, 0, 0, 0, 0, 0\}$$

Detecting Subtours

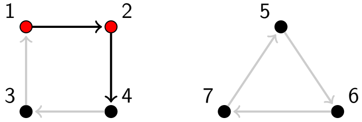
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2\}$$
$$Q = \{2\}$$
$$D = \{1, 1, 0, 0, 0, 0, 0\}$$

Detecting Subtours

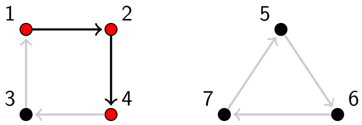
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2\}$$
$$Q = \{\}$$
$$D = \{1, 1, 0, 0, 0, 0, 0\}$$

Detecting Subtours

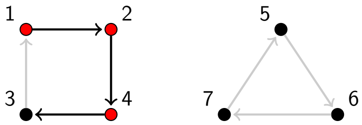
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2, 4\}$$
$$Q = \{4\}$$
$$D = \{1, 1, 0, 1, 0, 0, 0\}$$

Detecting Subtours

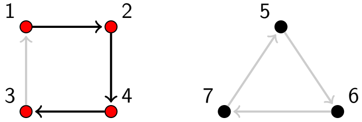
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2, 4\}$$
$$Q = \{\}$$
$$D = \{1, 1, 0, 1, 0, 0, 0\}$$

Detecting Subtours

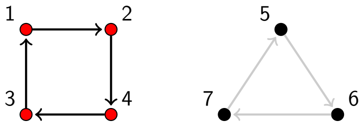
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2, 4, 3\}$$
$$Q = \{3\}$$
$$D = \{1, 1, 0, 1, 0, 0, 0\}$$

Detecting Subtours

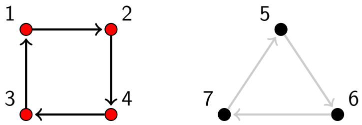
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\}$$
$$K = \{1, 2, 4, 3\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

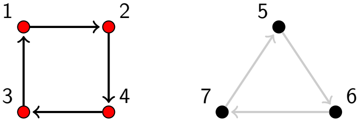
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\mathbf{1}, 2, 3, 4, 5, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \mathbf{2}, 3, 4, 5, 6, 7\}$$

$$C = \{\{1, 2, 4, 3\}\}$$

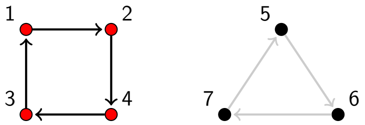
$$K = \{\}$$

$$Q = \{\}$$

$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

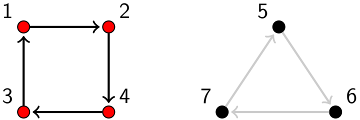
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \mathbf{3}, 4, 5, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

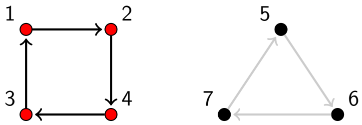
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \mathbf{4}, 5, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

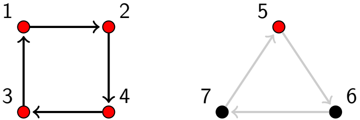
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, 5, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 0, 0, 0\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \mathbf{5}, 6, 7\}$$

$$C = \{\{1, 2, 4, 3\}\}$$

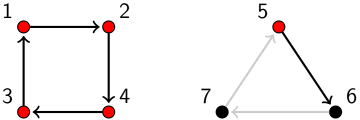
$$K = \{5\}$$

$$Q = \{5\}$$

$$D = \{1, 1, 1, 1, 1, 0, 0\}$$

Detecting Subtours

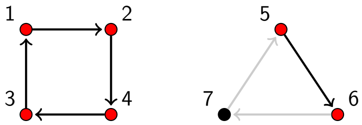
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{\mathbf{4}}, \mathbf{5}, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{5\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 1, 0, 0\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{\mathbf{4}}, \mathbf{5}, 6, 7\}$$

$$C = \{\{1, 2, 4, 3\}\}$$

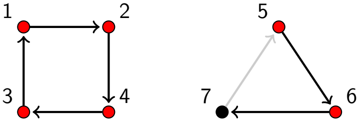
$$K = \{5, 6\}$$

$$Q = \{6\}$$

$$D = \{1, 1, 1, 1, 1, 1, 0\}$$

Detecting Subtours

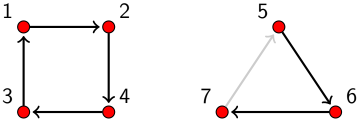
- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{\mathbf{4}}, \mathbf{5}, 6, 7\}$$
$$C = \{\{1, 2, 4, 3\}\}$$
$$K = \{5, 6\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 1, 1, 0\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \mathbf{5}, 6, 7\}$$

$$C = \{\{1, 2, 4, 3\}\}$$

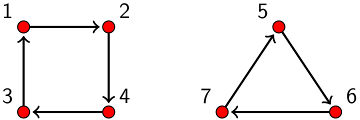
$$K = \{5, 6, 7\}$$

$$Q = \{7\}$$

$$D = \{1, 1, 1, 1, 1, 1, 1\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.



$$N = \{\cancel{\mathbf{1}}, \cancel{\mathbf{2}}, \cancel{\mathbf{3}}, \cancel{\mathbf{4}}, \mathbf{5}, 6, 7\}$$

$$C = \{\{1, 2, 4, 3\}\}$$

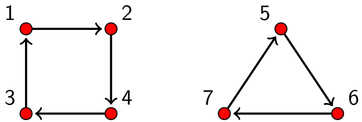
$$K = \{5, 6, 7\}$$

$$Q = \{\}$$

$$D = \{1, 1, 1, 1, 1, 1, 1\}$$

Detecting Subtours

- Given a solution current feasible integer solution $\bar{x}_{ij}$.

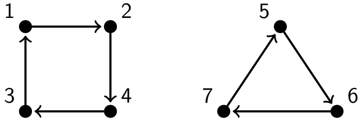


$$N = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6}, \cancel{7}\}$$
$$C = \{\{1, 2, 4, 3\}, \{5, 6, 7\}\}$$
$$K = \{\}$$
$$Q = \{\}$$
$$D = \{1, 1, 1, 1, 1, 1, 1\}$$

- This procedure allows us to identify all of the subtours at a current solution and add the cuts for each of them

Example

For the current solution

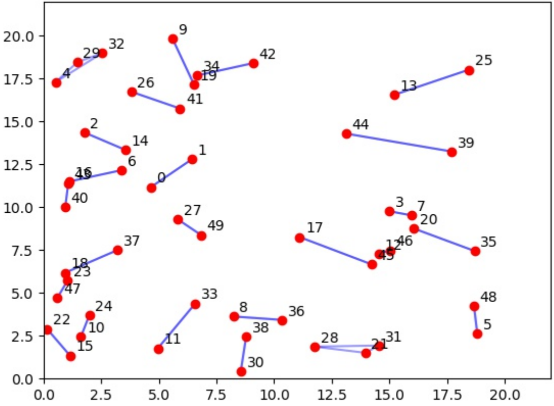


The corresponding subtour elimination cuts are

$$x_{12} + x_{21} + x_{13} + x_{31} + x_{14} + x_{41} + x_{23} + x_{32} + x_{24} + x_{42} + x_{34} + x_{43} \leq 3$$

$$x_{56} + x_{65} + x_{57} + x_{75} + x_{67} + x_{76} \leq 2$$

Visualization



Contents

Lazy Cuts

- We will implement our lazy cuts in a **callback function**
- Lazy cuts use different attributes than user cuts and occur at different points of the branch and bound tree
- The essential Gurobi attributes for user and lazy cuts:

Attribute	User Cuts	Lazy Cuts
parameters	Model.Params.PreCrush	Model.Params.lazyConstraints
where	GRB.Callback.MIPNODE	GRB.Callback.MIPSOL
current solution	Model.cbGetNodeRel()	Model.cbGetSolution()
add constraint	Model.cbCut()	Model.cbLazy()

Future Work

- We will now go to the Jupyter notebook to solve the TSP
- In a supplemental workshop we will cover some other techniques used solve or approximate solutions of the TSP
- These techniques include:
  1. Lagrangian Relaxation
  2. Approximation Algorithms
  3. Heuristics