Lecture 11: Travelling Salesman Problem

Junaid Hasan

Math 381: Discrete Math Modelling

Travelling Salesman Problem

- We ask the following question:
- Given a weighted graph, with non-negative weights, find a hamiltonian cycle of least total weight.
- Analogously, we may ask:
- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the original city?

Observation:

- Observe that the underlying graph G can be expanded to a complete graph
- if we take the weights of the extra edges to be “infinity” or a large enough quantity that the cycle will not select them.
- Complete graphs are easier to write a linear program for.

Integer Linear Program

- Given an edge (i, j) let $c_{ij} = c_{ji}$ we the weight of the edge.
- In case we want to consider directed graphs, then we can take $c_{ij} \neq c_{ji}$.
- Let x_{ij} be a binary variable, that records
- 1 if path goes from city i to city j.
- 0 otherwise.

Almost:

- minimize

$$\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}$$

- Given a vertex i, we go from i to another vertex (and only one) j, therefore for each i,

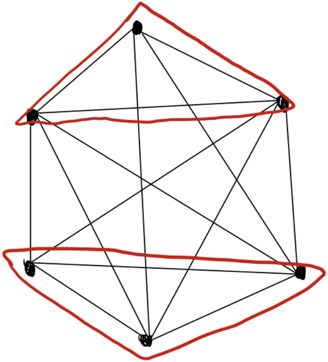
$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1$$

- Given a vertex j we must arrive at it from some vertex (and only one), therefore for each j,

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1$$

Question: Is this enough?

- Answer: Not quite!
- Reason: We may select more than one cycle. Because nothing prevents the program from that. Each smaller cycle is called a *subcycle*.
- Therefore we must avoid *subcycles*.



-

Tweaking the LP

- How do we solve the problem then?
- There are many ways, we discuss the Miller Tucker Zemlin formulation.
- Maybe we record the order in which we visit and make sure we do not do subcycles.
- Let us pick a starting vertex say 1.
- Let us keep a record of the order of visiting by a variable u_i for each vertex i.

contd..

- $u_1 = 1$ since we start at vertex 1.
- $2 \leq u_i \leq n$
- If path goes from i to j then $u_j = u_i + 1$, to enforce this the constraint:
- for $2 \leq i \neq j \leq n$

$$u_i - u_j + nx_{ij} \leq n - 1$$

Extra constraint proof

- To prove why for $2 \leq i \neq j \leq n$

$$u_i - u_j + nx_{ij} \leq n - 1$$

- works, suppose we have a subcycle K not containing vertex 1 and size k.
- For those vertices in the subcycle $x_{ij} = 1$ if the cycle goes from i to j, therefore we add the k constraints to get
- $$\sum_{(i,j) \in K} (u_i - u_j + n) \leq \sum_{(i,j) \in K} n - 1$$

Contd..

- Since K is a subcycle, $\sum_{(i,j)\in K} u_i - u_j = 0$.
- Therefore we obtain
- $$nk \leq (n-1)k$$
- which is impossible.
- Therefore we cannot have a subcycle not containing 1.

Contd..

- The constraint is feasible:
- If $x_{ij} = 0$, then the constraint $u_i - u_j + nx_{ij} \leq n - 1$ tells us that
- $$u_i - u_j \leq n - 1$$
  which is true because $2 \leq u_i, u_j \leq n$.
- If $x_{ij} = 1$, then for $2 \leq i, j \leq n$
  $$u_i - u_j + n \leq n - 1 \implies u_i + 1 \leq u_j$$
- The fact that u_i can be among $\{2, \dots, n\}$ forces $u_j = u_i + 1$.

TSP LP (MTZ Version)

- We state the LP for completeness:
- minimize

$$\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}$$

- for each i,

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1$$

- for each j,

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1$$

- $u_1 = 1,$ and $2 \leq u_i \leq n$ for $2 \leq i \leq n$ and
- for $2 \leq i, j \leq n$

$$u_i - u_j + n x_{ij} \leq n - 1$$

Other formulations

- There are other formulations as well.
- Another way to avoid subcycles is to ensure that for every proper subset Q of size $2 \leq |Q| \leq n-1$ of the vertices
- the sum

$$\sum_{i,j \in Q, i \neq j} x_{ij} \leq |Q| - 1$$

- In other words
- for every proper subset subset Q such that $2 \leq |Q| \leq n-1$

$$\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} \leq |Q| - 1$$

- It is known as Dantzig Fulkerson Johnson Formulation (DFJ). The caveat is that here we have $2^n - (n+2)$ subsets Q which enforce exponential number of constraints.

Applications

- practically everywhere
- logistics, planning,
- DNA sequencing, chip design etc.

Variants

- If the weights satisfy the triangle inequality

$$c_{AB} \leq c_{AC} + c_{CB}$$

- this is a *metric* TSP.
- We may consider other versions
- For example we may want asymmetry $c_{ij} \neq c_{ji}$
- Or we may allow certain vertices to be visited multiple times.