TRAVELLING SALESMAN PROBLEM

OPTIMIZATION METHODS I

Final Project

Report

Dikshant Joshi

Dikshant.joshi@unh.edu

DS 806.01 – Optimization Methods I

Final Project Report

Dikshant Joshi

Travelling Salesman Problem

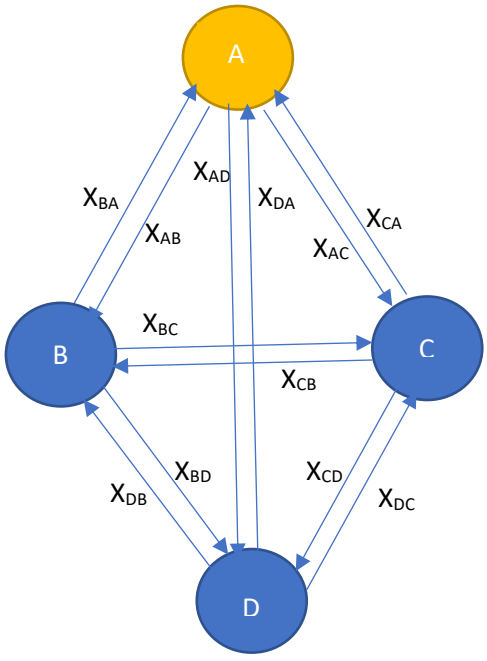
- Overview about the problem:

  The Travelling Salesman Problem (TSP) is a classic optimization problem that involves finding the shortest possible route that a salesman can take to visit a set of cities exactly once and return to the starting city. The TSP has many practical applications in fields such as logistics, transportation, and manufacturing. However, the TSP is known to be an NP-hard problem, meaning that there is no known algorithm that can solve it efficiently for large problem sizes.

  In this project, we aim to solve the TSP using Integer Linear Programming (ILP), a powerful optimization technique that can be used to solve a wide range of combinatorial optimization problems. ILP involves modelling the problem as a set of linear equations and inequalities, with integer variables representing the decision variables of the problem. The ILP model is then solved using specialized software that can find the optimal solution to the problem.

- Example:

  For Example: There are three cities A, B, C, D, and a logistics company has to deliver one product in each city. There are distances involved with edges between each of the cities. Based on the distances logistics company has to find shortest path that would cover all the cities and return back to the starting point. The constraint involved with it is that each city should be visited one and only once.



DS 806.01 – Optimization Methods I

Final Project Report

Dikshant Joshi

- Formulation of LP:

Decision Variables:

Let X_{ij} be the decision variable for the TSP, where X_{ij} is a binary variable that indicates whether the salesman travels directly from city i to city j. Therefore, if $X_{ij} = 1$, then the salesman travels from city i to city j, and if $X_{ij} = 0$, then the salesman does not travel directly from city i to city j.

Objective Function:

The objective function for the TSP is to minimize the total distance travelled by the salesman. Therefore, the objective function can be expressed as:

Minimize Z: $\sum_{i,j} C_{ij} \times X_{ij}$

where C_{ij} is the distance between cities i and j.

Minimize Z = $C_{AB}*X_{AB} + C_{AC}*X_{AC} + C_{AD}*X_{AD} + C_{BA}*X_{BA} + C_{CA}*X_{CA} + C_{DA}*X_{DA} + C_{DB}*X_{DB} + C_{BD}*X_{BD} + C_{DC}*X_{DC} + C_{CD}*X_{CD} + C_{BC}*X_{BC} + C_{CB}*X_{CB}$

Constraints:

The constraints for the TSP ensure that the salesman visits each city exactly once and returns to the starting city. The constraints can be expressed as follows:

- **The salesman must leave each city exactly once:**

  $\sum_j X_{ij}$ = 1 for all i

  ➔$X_{AB} + X_{AC} + X_{AD} = 1$, $X_{BA} + X_{BC} + X_{BD} = 1$, $X_{CA} + X_{CB} + X_{CD} = 1$, $X_{DA} + X_{DB} + X_{DC} = 1$

  This constraint ensures that the salesman leaves each city exactly once, as the sum of X_{ij} over all j for each i must be equal to 1.

- **The salesman must visit each city exactly once:**

  $\sum_i X_{ij}$ = 1 for all j

  ➔$X_{BA} + X_{CA} + X_{DA} = 1$, $X_{AB} + X_{CB} + X_{DB} = 1$, $X_{AC} + X_{BC} + X_{DC} = 1$, $X_{AD} + X_{BD} + X_{CD} = 1$

  This constraint ensures that the salesman enters each city exactly once, as the sum of Xij over all i for each j must be equal to 1.

DS 806.01 – Optimization Methods I

Final Project Report

Dikshant Joshi

- **The TSP should contain n number of edges:**

  $\sum_{i,j} X_{ij}$ <= n

  $X_{AB} + X_{AC} + X_{AD} + X_{BA} + X_{BC} + X_{BD} + X_{CA} + X_{CB} + X_{CD} + X_{DA} + X_{DB} + X_{DC} <= 4$

  This constraint ensures that the TSP covers all the cities. Because to connect n nodes completely in a cycle you need to have minimum n edges as we should come back to the point where we started to solve TSP.

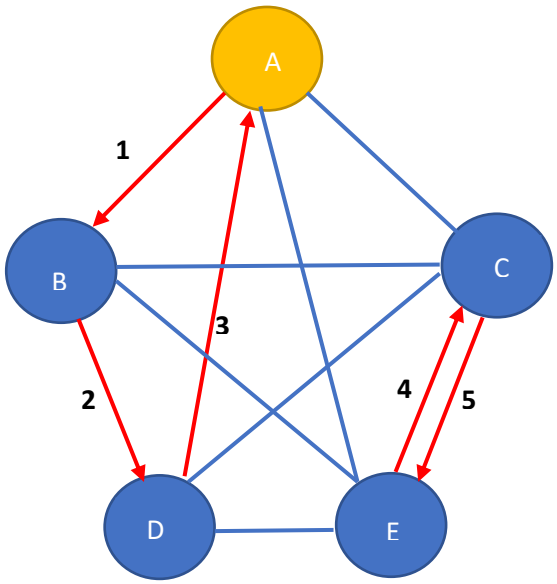
- **Subtour Elimination**

  For each subset of nodes with at-least two nodes we limit the maximum number of edges selected:

  $\sum_{i \in S, j \in S, i \neq j} X_{ij}$ <= |S|-1 $\forall$ S $\subset$ V , |S| >= 2 , where V is set of all nodes and S is proper subset of V.

  So for instance lets suppose we have cities A,B,C,D,E & we have a subtour S = {A,B,D} and other Subtour S={C,D}.

  If we look at the constraint 1, 2 & 3 they are satisfied ensuring that there should be one outgoing & one incoming edge from each node and sum of selected number of edges is n (5). But still the condition is not satisfied as we are not able to cover all the cities in a cycle.



Our 4 constraint avoids this by eliminating all possible subtours. In this case where S={A,B,D} , |S| = 3. And so the constraint which will get imposed on it will be

$\sum_{i\in S, j\in S, i\neq j} X_{ij}$ <= |S|-1 ∀ S ⊂ V , |S| >= 2

➔$X_{AB} + X_{BD} + X_{DA}$ <= 3-1

➔ $X_{AB} + X_{BD} + X_{DA}$ <= 2

Hence out of the given three edges at maximum only 2 edges will be selected eliminating the third edge and, thus eliminating the subtour.

Now where S={C,D}, |S|=2. And so the constraint which will get imposed on it will be

$\sum_{i\in S, j\in S, i\neq j} X_{ij}$ <= |S|-1 ∀ S ⊂ V , |S| >= 2

➔$X_{CD} + X_{DC}$ <= 2-1

➔ $X_{CD} + X_{DC}$ <= 1

Hence out of the given edges only one will be selected eliminating the second edge and, thus eliminating the subtour.

This way this constraint imposed on all the subset of nodes whose cardinality>=2 will eliminate all the possible subtours.

- Alternatively for subtour elimination MTZ(Miller-Tucker-Zemlin) constraint can also be used. This constraint is as given below:

  $U_i - U_j + N*X_{ij} <= N-1$ ∀ i,j : i≠j , j≠1

  Where N is number of nodes and U_i variable is the sequence of visit of node i. So for instance lets take above example of subtour {A,B,D}. For this subtour to exist :

  $X_{AB} = 1$, $X_{BD}=1$, $X_{DA}=1$

  Now the MTZ constraint will check the conditions. Lets suppose $U_A = 1$.

  - $U_A - U_B + N*X_{AB} <= N-1$
  - ⇨ $U_A - U_B + N <= N-1$
  - ⇨ $U_A - U_B + 1 <= 0$
  - ⇨ $U_A + 1 <= U_B$, $U_B = 2$,
  - $U_B - U_D + N*X_{BD} <= N-1$
  - ⇨ $U_B - U_D + N <= N-1$
  - ⇨ $U_B - U_D + 1 <= 0$
  - ⇨ $U_B + 1 <= U_D$, $U_C = 3$,
  - $U_D - U_A + N*X_{CD} <= N-1$
  - ⇨ $U_D - U_A + N <= N-1$
  - ⇨ $U_D - U_A + 1 <= 0$
  - ⇨ $U_D + 1 <= U_A$, which will contradict with $U_A = 1$ and hence will eliminate $X_{BA} = 1$ thus eliminating a subtour.

These constraints ensure that the salesman visits each city exactly once, returns to the starting city, and travels in a cycle that does not repeat any cities. Together with the objective function, they form a complete model for the TSP optimization problem using ILP.

- Modelling the problem in AMPL:

Model File

```
reset;

set nodes;
set arcs within nodes cross nodes;
param Dist{arcs};
var X{arcs} binary>=0;
var U {i in 2..card(nodes)} >= 2,<=card(nodes);
minimize total_distance:
 sum{i in nodes, j in nodes} Dist[i,j]*X[i,j];

subject to one_outgoing{k in nodes}:
    sum{i in nodes} X[i,k]= 1;

subject to one_incoming{k in nodes}:
    sum{j in nodes} X[k,j] = 1;

subject to no_subtour{k in nodes, j in nodes: j > 1 and k > 1}:
    U[j] - U[k] + card(nodes)*X[j,k] <= card(nodes) - 1;

subject to number_ofEdges:
    sum{(i,j) in arcs}X[i,j] <=card(nodes);
```

Data File:

```
set nodes:= 1 2 3 4 5;

set arcs:
    1 2 3 4 5:=
1   + + + + +
2   + + + + +
3   + + + + +
4   + + + + +
5   + + + + +;

param Dist:
  1 2 3 4 5:=
 1 0 120 220 150 210
 2 120 0 80 110 130
 3 220 80 0 160 185
 4 150 110 160 0 190
 5 210 130 185 190 0
;
```

Output:

```
ampl: model 'C:\Masters\Unh\Optimization Method\class
CPLEX 20.1.0.0: sensitivity
CPLEX 20.1.0.0: optimal integer solution; objective 7
27 MIP simplex iterations
0 branch-and-bound nodes

suffix up OUT;
suffix down OUT;
suffix current OUT;
X :=
        1    2    3    4    5
    1   0    0    0    1    0
    2   1    0    0    0    0
    3   0    1    0    0    0
    4   0    0    0    0    1
    5   0    0    1    0    0
U [*] :=
2  5
3  4
4  2
5  3
;

total_distance = 725

ampl:
```