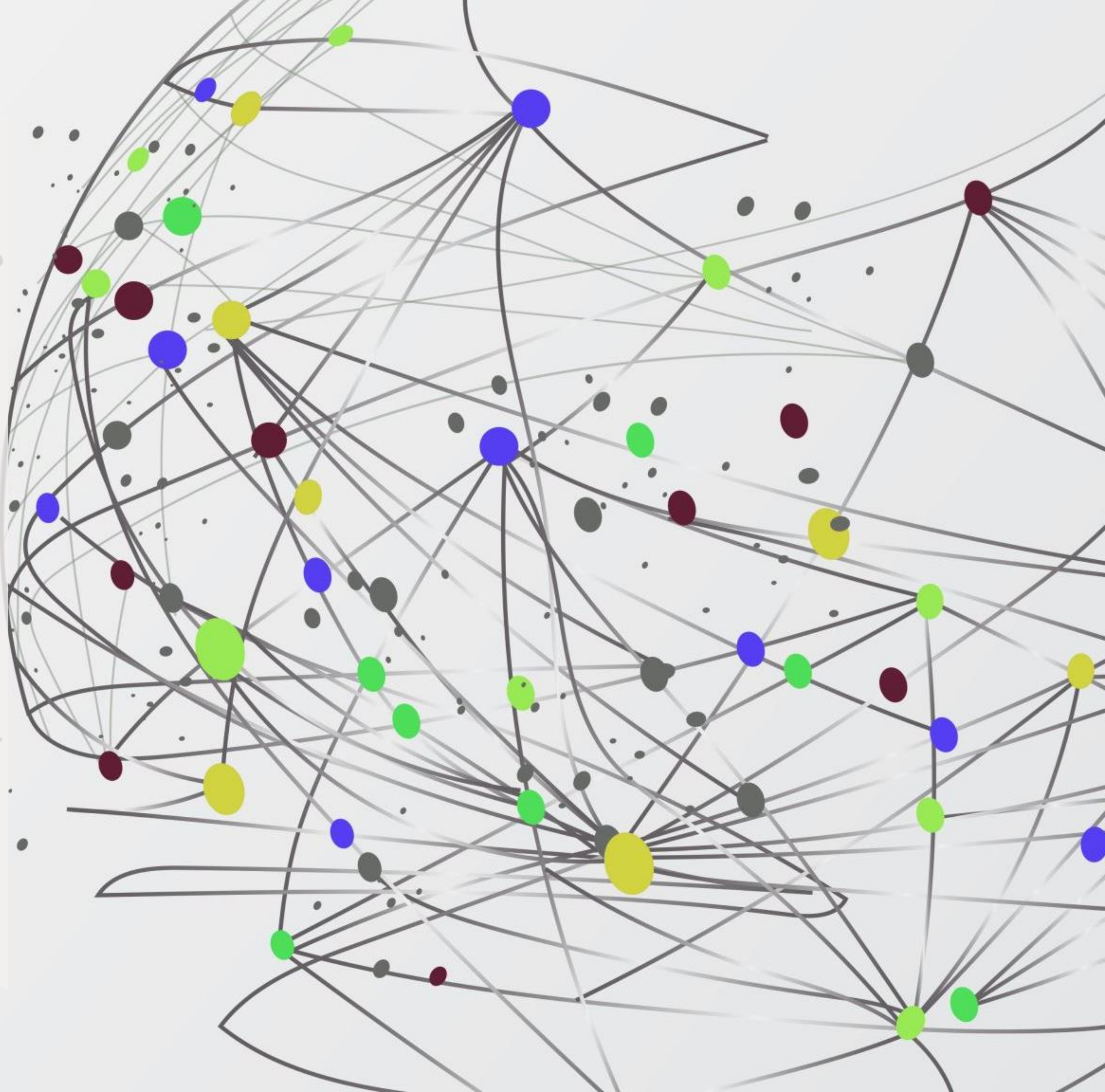
TRAVELLING SALESMAN PROBLEM

OPTIMIZATION METHODS - I

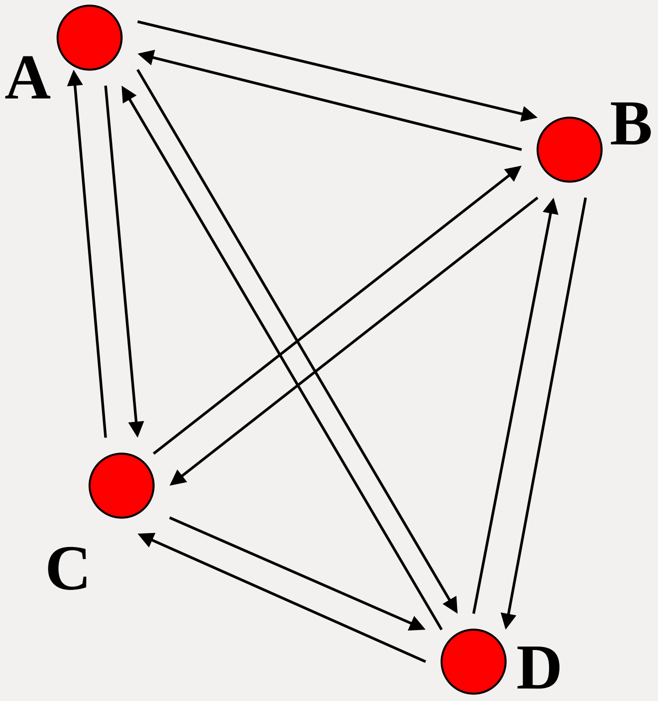
-Dikshant Joshi



Agenda

- Overview
- Application of TSP
- Problem Statement
- Formulation of LP
  - Subtour Elimination
- Modelling using AMPL
- Results

OVERVIEW



- The Travelling Salesman Problem (TSP) is a classic optimization problem.
- It 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.

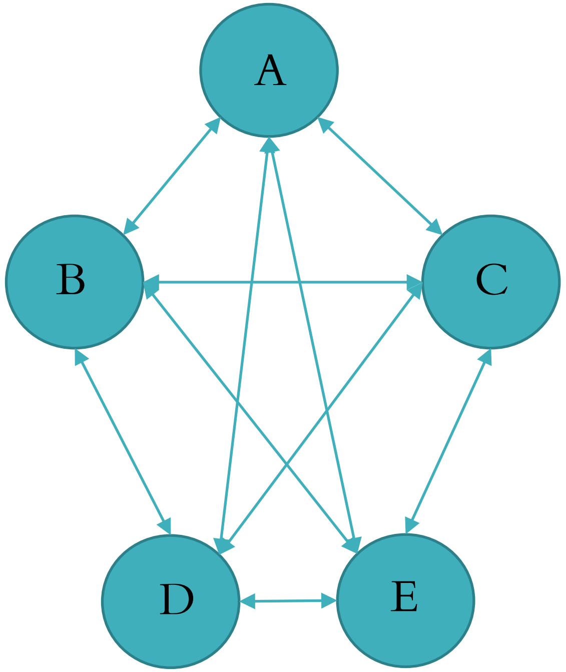
APPLICATION OF TSP

- Planning Delivery Routes
- Optimizing Manufacturing Processes
- Job Processing
- School Bus Routing

Theorem: (Karp, 1972) TSP is NP-Complete. (That is, every other hard problem is reducible to TSP.)

PROBLEM STATEMENT

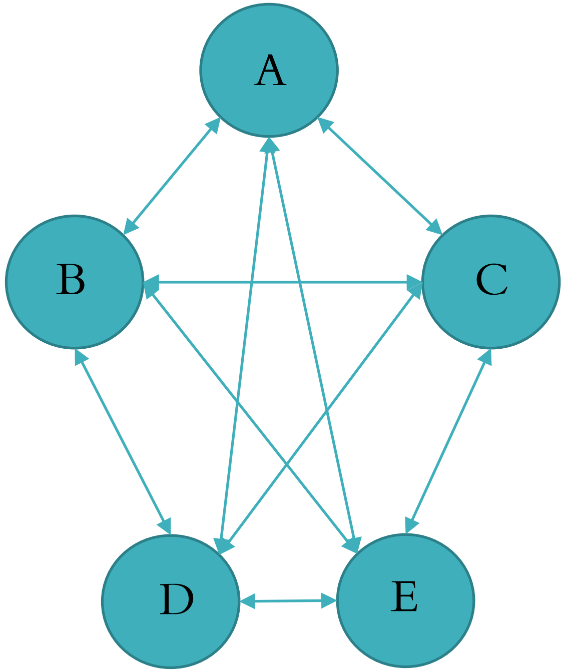
- There are 5 cities A,B,C,D,E and a logistics company must deliver one product in each city.
- There are distances involved with edges between each of the cities.
- Based on the distance logistics company must find shortest path that would cover all the cities and return to the starting city.



	A	B	C	D	E
A	0	120	220	150	210
B	120	0	80	110	130
C	220	90	0	160	185
D	150	110	160	0	190
E	210	130	185	190	0

? PROBLEM STATEMENT

- There are 5 cities A,B,C,D,E and a logistics company must deliver one product in each city.
- There are distances involved with edges between each of the cities.
- Based on the distance logistics company must find shortest path that would cover all the cities and return to the starting city.



	A	B	C	D	E
A	0	120	220	150	210
B	120	0	80	110	130
C	220	90	0	160	185
D	150	110	160	0	190
E	210	130	185	190	0

FORMULATION OF LP

- **Decision Variables**
  - » X_{ij} : a binary variable that indicates whether the salesman travels directly from city i to city j (1 if yes, 0 if not).

- **Objective Function:**
  - » Minimize Z: $\sum_{i,j} C_{ij} \times X_{ij}$ $\qquad ij \in E,$

Where E is the set of all edges and C is distance between city i & j .

	A	B	C	D	E
A	0	120	220	150	210
B	120	0	80	110	130
C	220	90	0	160	185
D	150	110	160	0	190
E	210	130	185	190	0

FORMULATION OF LP

- **Constraints**
  - Salesman must visit each city exactly once:

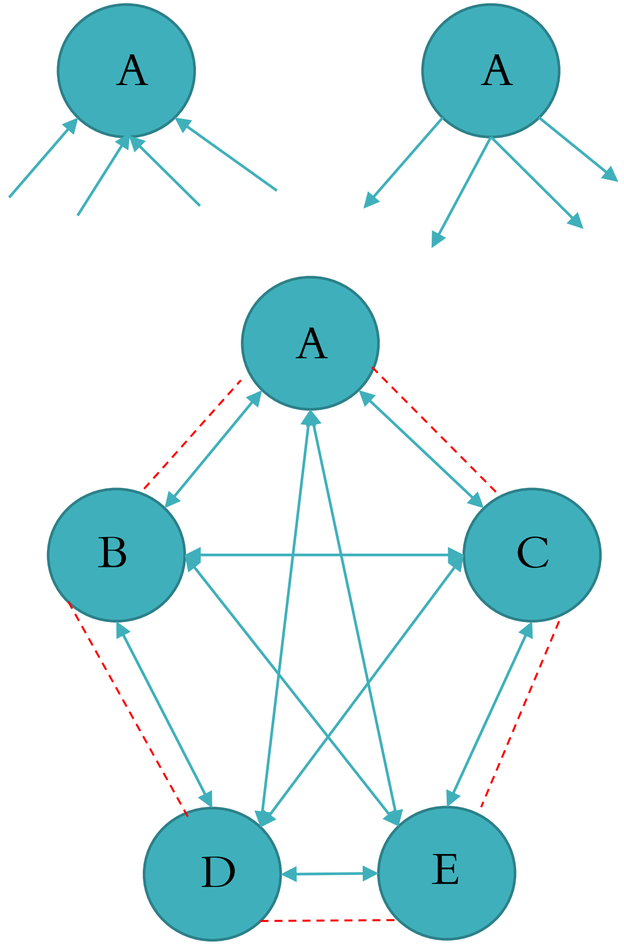
    $\sum_i X_{ij}$ = 1 for all j

  - Salesman must leave each city exactly once:

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

  - The optimum route should contain at-most n edges:

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



FORMULATION OF LP

- Constraints
  - **Subtour Elimination:**

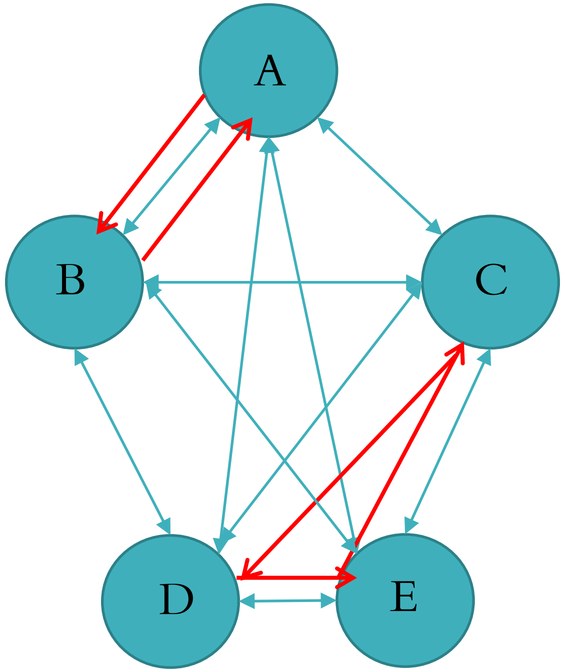
Method 1

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

Method 2 (MTZ – Miller Tucker Zemlin)

$$U_i - Uj + N * Xij <= N - 1 \; \forall i, j : \; i \neq 1, j \neq 1, \; 2 <= U_i <= N \text{ given } U_1 = 1$$

where N is the number of Nodes and U_i variable is the sequence of visit of node i.



Subtour Elimination

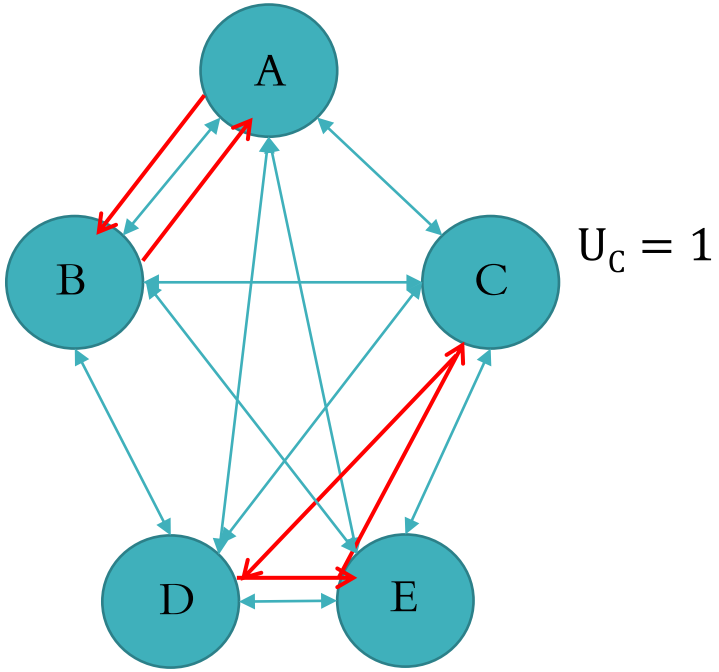
- $U_i - Uj + N * Xij <= N - 1 \; \forall \; i,j : i \neq 1 , j \neq 1$

  $U_C - UD + 5 * XCD \leq 4$

  $U_C - UD + 5 * 1 \leq 4$

  $U_C - UD + 1 \leq 0$

  $U_C + 1 \leq U_D = 2$



Subtour Elimination

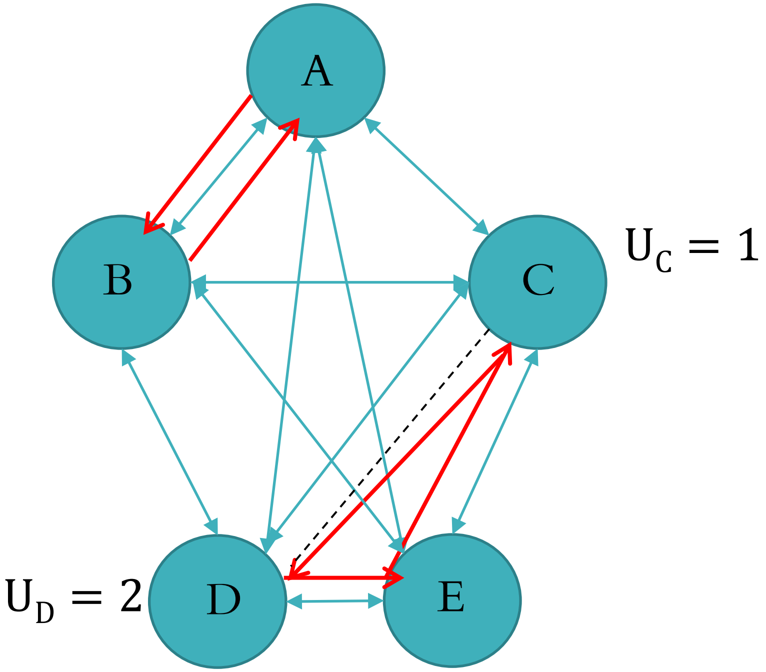
- $U_i - Uj + N * Xij <= N - 1 \; \forall \, i, j : \; i \neq 1 , j \neq 1$

  $U_C - UD + 5 * XCD \leq 4$

  $U_C - UD + 5 * 1 \leq 4$

  $U_C - UD + 1 \leq 0$

  $U_C + 1 \leq U_D = 2$



Subtour Elimination

- $U_i - Uj + N * Xij <= N - 1 \; \forall \, i, j : \; i \neq 1, j \neq 1$

$U_C - UD + 5 * XCD \leq 4$

$U_C - UD + 5 * 1 \leq 4$

$U_C - UD + 1 \leq 0$

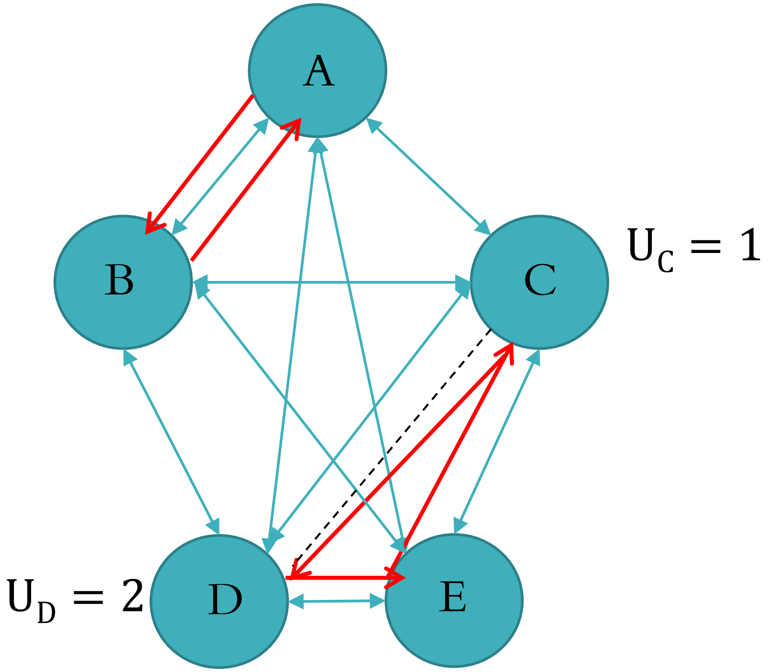
$U_C + 1 \leq U_D = 2$

$U_D - UE + 5 * XDE \leq 4$

$U_D - UE + 5 * 1 \leq 4$

$U_D - UE + 1 \leq 0$

$U_D + 1 \leq U_E = 3$



Subtour Elimination

- $U_i - Uj + N * Xij <= N - 1 \; \forall \, i, j : i \neq 1, j \neq 1$

  $U_C - UD + 5 * XCD \leq 4$

  $U_C - UD + 5 * 1 \leq 4$

  $U_C - UD + 1 \leq 0$

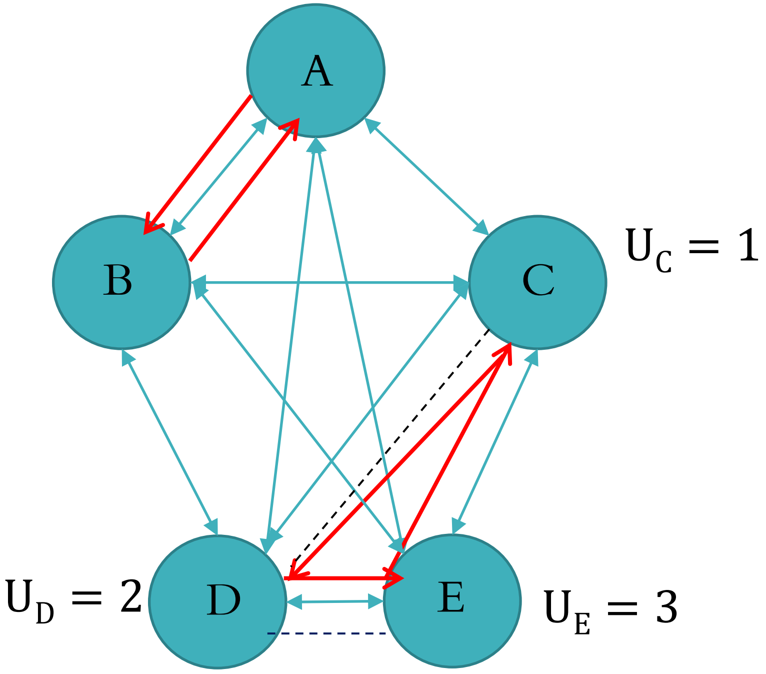
  $U_C + 1 \leq U_D = 2$

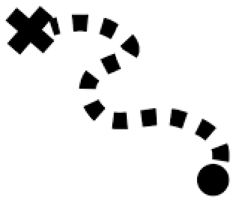
  $U_D - UE + 5 * XDE \leq 4$

  $U_D - UE + 5 * 1 \leq 4$

  $U_D - UE + 1 \leq 0$

  $U_D + 1 \leq U_E = 3$





Subtour Elimination

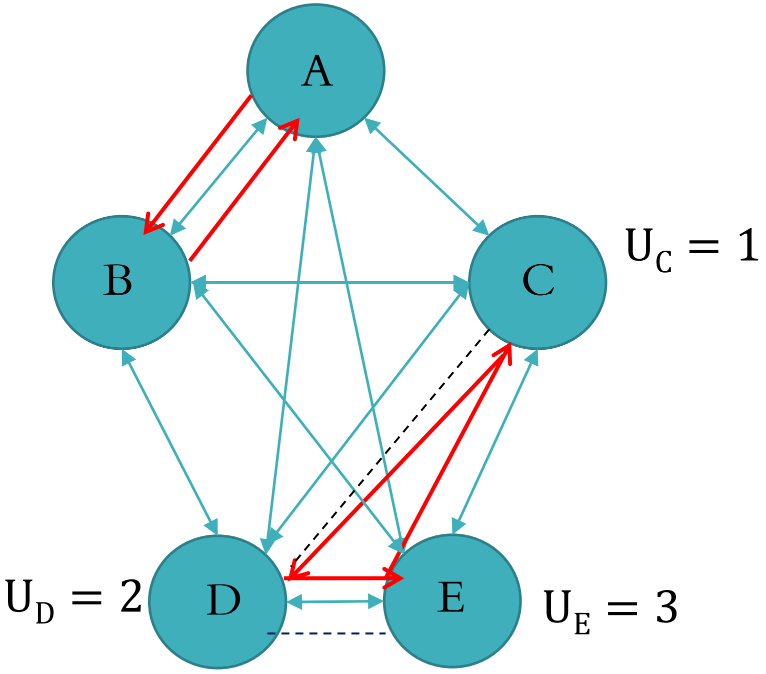
- $U_i - Uj + N * Xij <= N - 1 \; \forall \, i, j : \; i \neq 1, j \neq 1$

$U_C - UD + 5 * XCD \leq 4$
$U_C - UD + 5 * 1 \leq 4$
$U_C - UD + 1 \leq 0$
$U_C + 1 \leq U_D = 2$

$U_D - UE + 5 * XDE \leq 4$
$U_D - UE + 5 * 1 \leq 4$
$U_D - UE + 1 \leq 0$
$U_D + 1 \leq U_E = 3$

$U_E - UC + 5 * XEC \leq 4$
$U_E - UC + 5 * 1 \leq 4$
$U_E - UC + 1 \leq 0$
$U_E + 1 \leq U_C = 4$

Which contradicts with $U_C=1$



Subtour Elimination

- $U_i - Uj + N * Xij <= N - 1 \; \forall \, i, j : \; i \neq 1, j \neq 1$

  $U_C - UD + 5 * XCD \leq 4$

  $U_C - UD + 5 * 1 \leq 4$

  $U_C - UD + 1 \leq 0$

  $U_C + 1 \leq U_D = 2$

  $U_D - UE + 5 * XDE \leq 4$

  $U_D - UE + 5 * 1 \leq 4$

  $U_D - UE + 1 \leq 0$

  $U_D + 1 \leq U_E = 3$

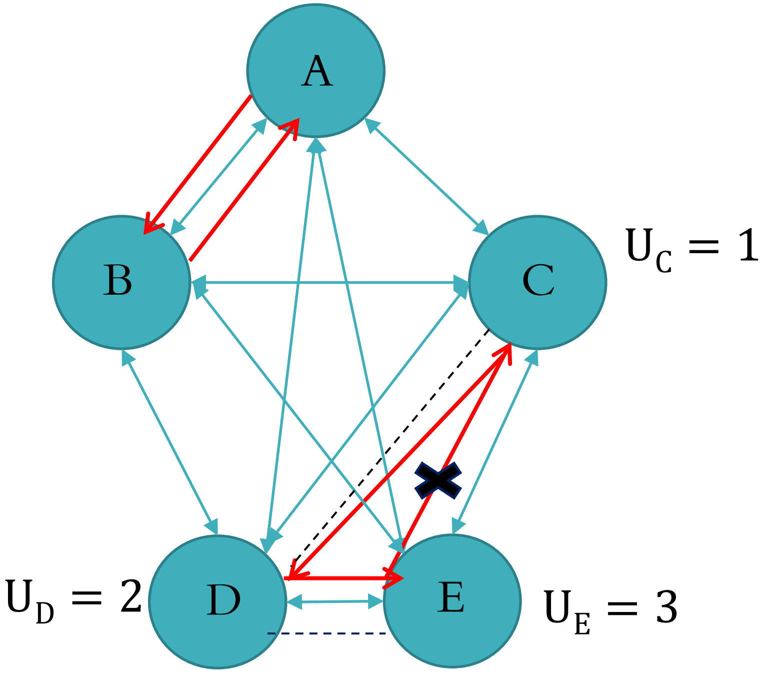
  $U_E - UC + 5 * XEC \leq 4$

  $U_E - UC + 5 * 1 \leq 4$

  $U_E - UC + 1 \leq 0$

  $U_E + 1 \leq U_C = 4$

  Which contradicts with $U_C=1$



MODELLING USING 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
;
```

RESULT

```
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:
```

The Optimum Route for the problem is :
1 (A) → 4 (D) → 5 (E) → 3 (C) → 2 (B) → 1 (A)

The minimized distance : 725


THANK YOU
QUESTIONS ?