22nd November 2022

ANM Term Project

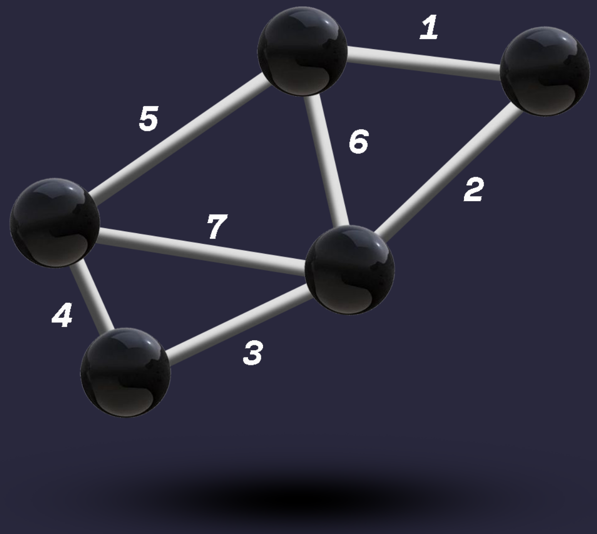
Finding the shortest path connecting all the nodes in a graph such that every node is visited only once

Chinmay Sahoo | Chinmay Ratnaparkhe

/TABLE OF CONTENTS

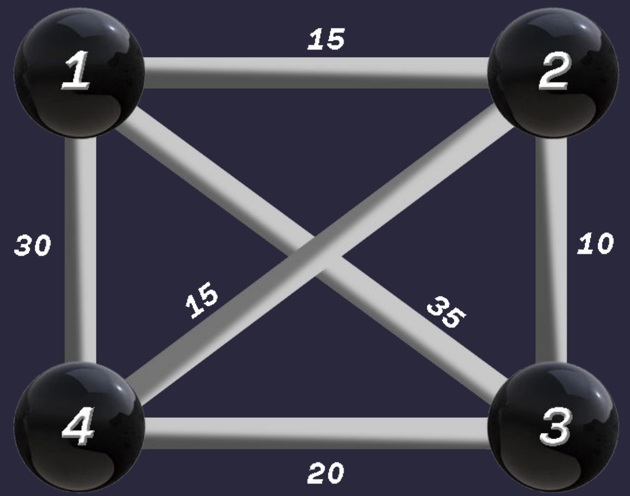
/01 Problem Statement

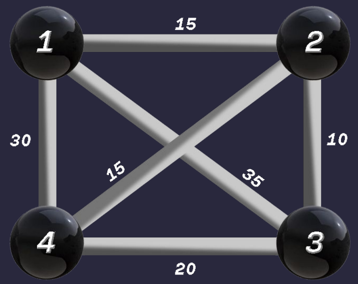
Finding the shortest possible path which connects all the nodes in a graph



/02 Mathematical Model

We will use this demo graph network for our analysis →





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

$$Z = \min(15x_{12} + 15x_{21} + 10x_{23} + 10x_{32} + 15x_{24} + 15x_{42} + 35x_{13} + 35x_{31} + 20x_{34} + 20x_{43} + 30x_{41} + 30x_{14})$$

Constraints :

Entry Exit Constraints

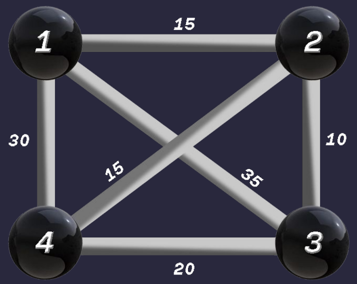
1) $x_{12} + x_{13} + x_{14} = 1$
2) $x_{21} + x_{31} + x_{41} = 1$
3) $x_{21} + x_{24} + x_{23} = 1$
4) $x_{12} + x_{42} + x_{32} = 1$
5) $x_{34} + x_{31} + x_{32} = 1$
6) $x_{23} + x_{13} + x_{43} = 1$
7) $x_{14} + x_{24} + x_{34} = 1$
8) $x_{41} + x_{42} + x_{43} = 1$

Avoiding 2 Node Subroutes Constraints

9) $x_{12} + x_{21} \leq 1$
10) $x_{23} + x_{32} \leq 1$
11) $x_{34} + x_{43} \leq 1$
12) $x_{14} + x_{41} \leq 1$
13) $x_{13} + x_{31} \leq 1$
14) $x_{24} + x_{42} \leq 1$

Avoiding 3 Node Subroutes Constraints

15) $x_{12} + x_{21} + x_{23} + x_{32} + x_{13} + x_{31} \leq 2$
16) $x_{12} + x_{21} + x_{14} + x_{41} + x_{24} + x_{42} \leq 2$
17) $x_{23} + x_{32} + x_{34} + x_{43} + x_{24} + x_{42} \leq 2$
18) $x_{13} + x_{31} + x_{34} + x_{43} + x_{14} + x_{41} \leq 2$



$$Z = \min \Sigma d_{ij} x_{ij} \quad \forall\ (i,j) \in E$$

Constraints :

Entry Exit Constraints :-

$\forall\ k \in V$

Only one incoming arc :

$$\sum_{i \in V, i \neq k} x_{ik} = 1$$

(n constraints)

Only one outgoing arc :

$$\sum_{j \in V, j \neq k} x_{kj} = 1$$

(n constraint)

Avoiding Subroutes Constraints

$$\sum_{i \in S, j \in S, i \neq j} x_{ij} \leq |S| - 1 \quad \forall S \subsetneq V, |S| \geq 2.$$

(2^n-n-2 constraints)

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

V : set of all nodes of the graph

$n = |V|$

E : set of all edges of the graph

/03

Brute Force Approach

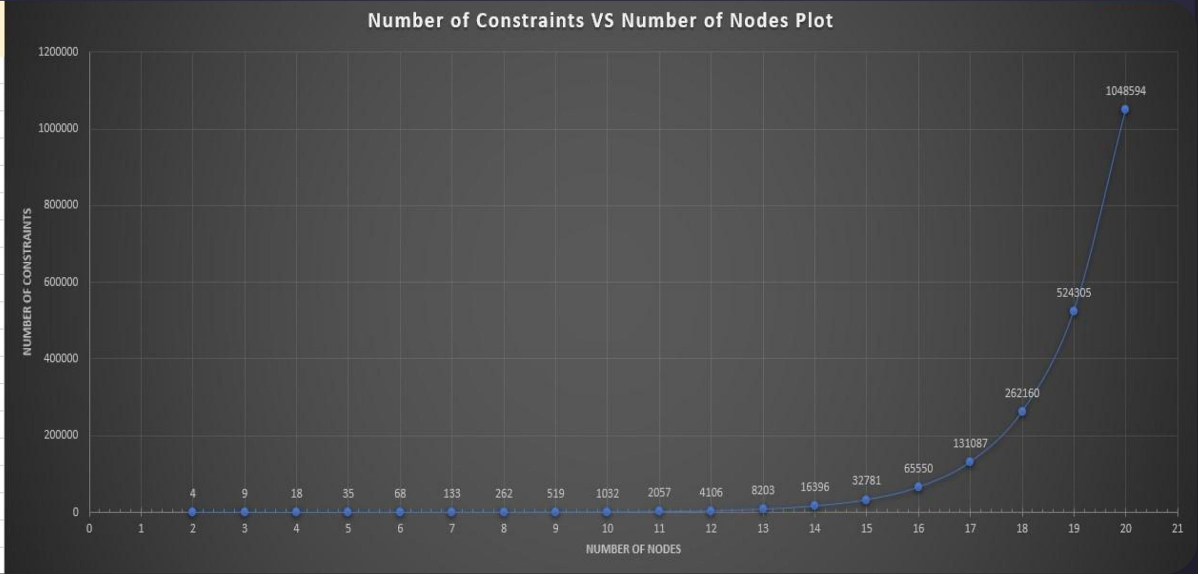
One way to solve the optimization problem:

Trying all possible combinations of x_{ij}'s

Total Number of Constraints:

$$2^n + n - 2$$

Number of Nodes	Number of Constraints
2	4
3	9
4	18
5	35
6	68
7	133
8	262
9	519
10	1032
11	2057
12	4106
13	8203
14	16396
15	32781
16	65550
17	131087
18	262160
19	524305
20	1048594



Total Number of Possible Solutions :

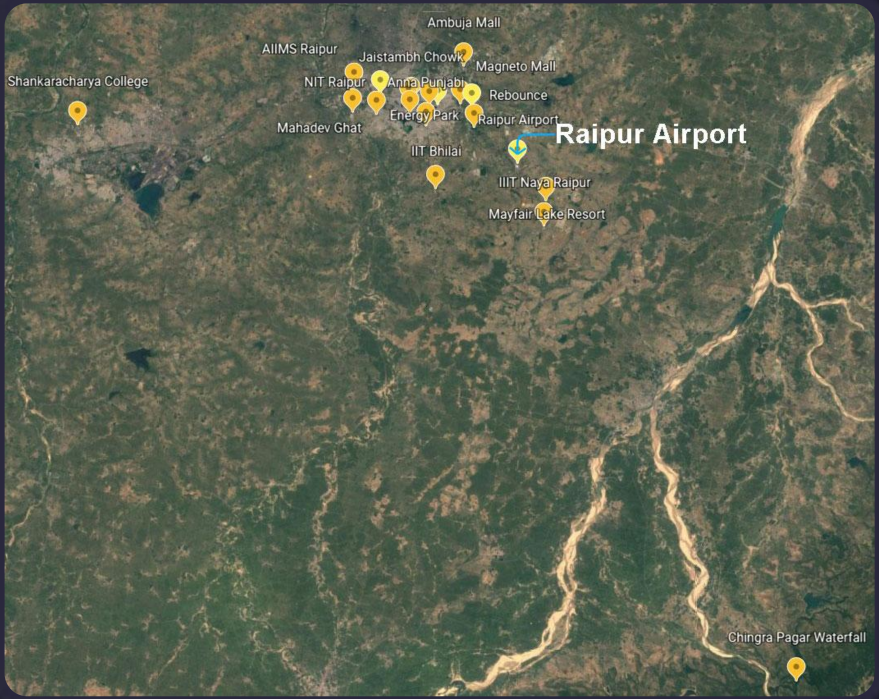
$$2^{n(n-1)}$$

Number of Nodes	Number of Variables	Number of Possible Solutions
2	2	4
3	6	64
4	12	4096
5	20	1048576
6	30	1073741824
7	42	4398046511104
8	56	72057594037927900
9	72	4722366482869650000000
10	90	1237940039285380000000000000
11	110	1298074214633710000000000000000000
12	132	5444517870735020000000000000000000000000
13	156	91343852333181400000000000000000000000000000000
14	182	6129982163463560000000000000000000000000000000000000000
15	210	1645504557321210000000000000000000000000000000000000000000000000
16	240	1766847064778380000000000000000000000000000000000000000000000000000000000
17	272	7588550360256750000000000000000000000000000000000000000000000000000000000000000000
18	306	130370302485407000000000000000000000000000000000000000000000000000000000000000000000000000000
19	342	8958978968711220000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
20	380	2462625387274650000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

/04 Christofides Algorithm

Case study of optimum way to visit some of the tourist locations in Raipur

Sr No.	Place
0	Marine Drive
1	IIT Bhilai
2	Airport
3	NIT Raipur
4	IIIT Naya Raipur
5	AIIMS Raipur
6	Magneto Mall
7	Ambuja Mall
8	Shankaracharya College
9	Jaistamvh Chowk
10	Phool Chowk
11	Energy Park
12	Budha Talab
13	Wonderland Park
14	Anna Punjabi
15	Mahadev Ghat
16	Rebounce
17	Chingra Waterfall
18	Mayfair Resort
19	Chhattisgarh Club



Airport	⟷	IIT Bhilai	: ₹192
Airport	⟷	Chingra Pagar Waterfall	: ₹1725
Airport	⟷	Mayfair Resort	: ₹528
Airport	⟷	Magneto Mall	: ₹185
Marine Drive	⟷	IIT Bhilai	: ₹211
Marine Drive	⟷	Ambuja Mall	: ₹113
Marine Drive	⟷	Energy Park	: ₹140
Marine Drive	⟷	Mahadev Ghat	: ₹207
Rebounce	⟷	AIIMS Raipur	: ₹308
Rebounce	⟷	Airport	: ₹177
Rebounce	⟷	Magneto Mall	: ₹46
Rebounce	⟷	Budha Talab	: ₹144

1. Minimum Spanning Tree
2. Odd Degree Nodes
3. Minimum Cost Perfect Matching
4. Make Eulerian Graph
5. Find Eulerian Cycle
6. Remove Repeated Nodes

1) Brute force on the demo graph while following the constraints justified gets us a minimum Z of **75** for the network.
2) While using Christofides Algorithm on Demo Graph gets us a minimum of **85**!
3) On our case study, the most optimum way to travel the required locations is -

Airport -> Rebounce -> Magneto Mall -> Ambuja Mall -> Marine Drive -> IIIT Naya Raipur -> Mayfair Resort -> Chingra Waterfall -> Shankaracharya College -> IIT Bhilai -> Anna Punjabi -> Chhattisgarh Club -> Budha Talab -> Jaistambh Chowk -> Phool Chowk -> NIT Raipur -> Mahadev Ghat -> Wonderland Park -> AIIMS Raipur -> Energy Park -> Airport

And the cost for the complete tour is **₹6587.0**

This was obtained using Christofides Algorithm with a computational time of a few seconds. Using Brute Force analysis here would get us in a computational loop of 10^8 Years!

1) **Brute Force Approach:**
As we increase the number of nodes in a network, the constraints and the number of possible solutions increase exponentially - Although this gives us a solution with unparalleled accuracy, the time taken for the same makes the method practically impractical.

2) **Christofides Algorithm:**
Although we have a high rate of error here (50%), compared to the time taken in computation makes this the best possible method to calculate the solution! As a future scope of improvement on the method, there can be efforts on error minimization, which would be the mathematical side of this.

Research Papers & Published articles :

1) Supply chain optimization under risk and uncertainty: A case study for high-end server manufacturing
2) Sustainable supply chain optimisation: An industrial case study
3) Nexus TSP approach

YouTube Video References :

1) Traveling Salesman Problem | Dynamic Programming
2) TSP Christofides algorithm
3) Integer Programming Problems
4) Integer Programming: Traveling salesperson problem
5) Eulerian Graph Theory

Coding References :

1) NetworkX documentation
2) Pandas documentation
3) Numpy documentation

Standard Pricing provided by **Uber** & **Ola**
Geographical Data by **Google Earth**