Introduction to GIS Methods in Economics

Giorgio Chiovelli and Sebastian Hohmann

March 18, 2020

Overview

The plan for today

Network Analysis in GIS

- Introduction to Network
- Solving Best Route problem in QNEAT
- Djkastra Algorithm

Network in Applied Economics

- Storeygard, 2016
- Replication of Donaldson and Hornbeck, 2015

What is a Network in GIS

- GIS networks represent routes upon which people and goods can travel.
  - interconnected lines (edges)
  - intersections (junctions)
- The object traversing the network follows the edges, and junctions appear when at least two edges intersect.
- Junctions and edges can have certain attributes increasing the cost of traveling in the network (impedance)
- Networks are either directed or undirected

Types of networks in GIS

- Utility networks
  - water mains, sewage lines, etc
- **Transportation networks**
  - roads, railroads, etc
- Networks based on social connections

Common Appications in GIS Networks

- **Shortest Path.** Finding the shortest path between two points
  - Google Map and Waze
- **Traveling Salesman.** Reaching every point in a network in the most efficient way possible
  - UPS uses a traveling algorithm (no left turns!)
- **Network Partition.** Dividing up of regions in a network to zones or subcategories.
  - Optimal location of fire stations

Working with Network in QGIS

- Prepare your network
  - Clean your line (roads, railways) shapefile
  - Set your cost parameter, if needed. e.g. Railways faster than roads
- Solve for the best route problem (Dijkstra's algorithm)

Working with Network in QGIS

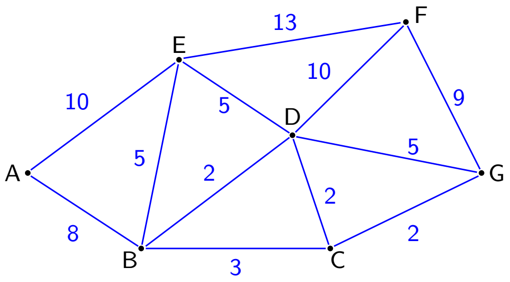
- QNEAT plugin [installable from the Plugins manager]
- QNEAT uses Dijkstra's Algorithm for two main application.
  - **O(rigin)D(estination) Matrix**.
    OD Matrix is ideal for calculation of big NxN matrixes as it does not draw the problem solution graphically. It visualizes the results of the Dijkstra's algorithm as straight lines.

    Typical application: Compute shortest route for each dyad of US county.
  - **Shortest Path Algorithms.**
    To be chosen when there is the need to produce maps showing the solution to a particular "shortest path problem".

    Typical application: Show the shortest route linking from Providence to Boston.

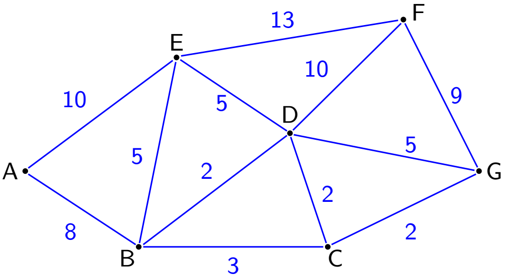
Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.



Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

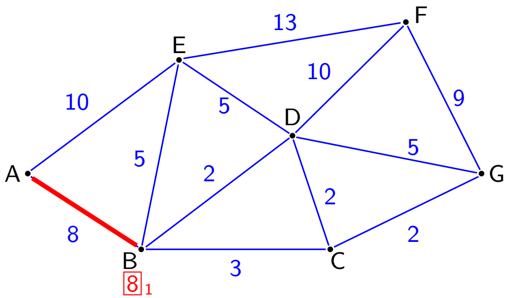


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

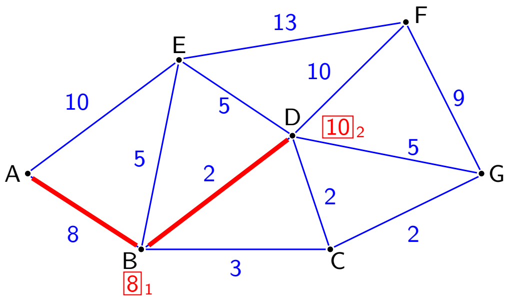


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

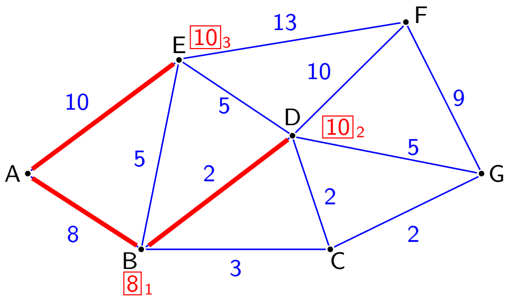


Algorithm

1. pick unvisited vertex with lowest distance (first vertex $= 0$)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

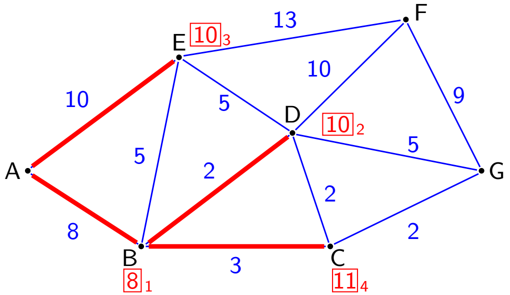


Algorithm

1. pick unvisited vertex with lowest distance (first vertex $= 0$)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

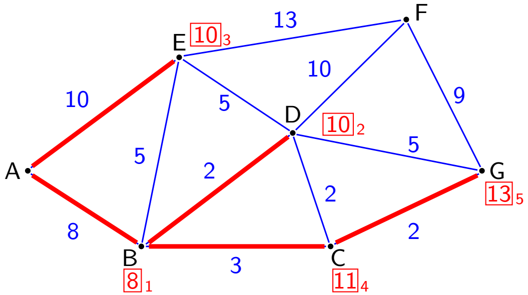


Algorithm

1. pick unvisited vertex with lowest distance (first vertex $= 0$)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Aim: Wish to travel $A \to F$ along **shortest path**.

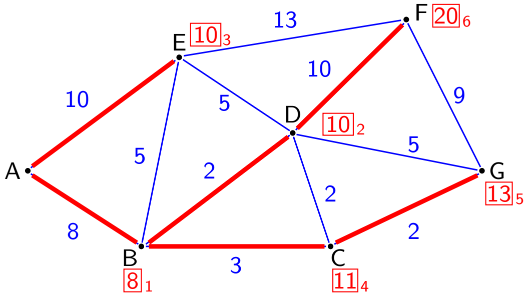


Algorithm

1. pick unvisited vertex with lowest distance (first vertex = 0)
2. calculate distance through each unvisited neighbour
3. pick unvisited neighbour with lowest distance as new current vertex
4. a vertex counts as visited once done with all its neighbours

Dijkstra's algorithm

Shortest path: *ABDF*, **search steps** 6, **distance** 20.



Algorithm

1 pick unvisited vertex with lowest distance (first vertex = 0)

2 calculate distance through each unvisited neighbour

3 pick unvisited neighbour with lowest distance as new current vertex

4 a vertex counts as visited once done with all its neighbours

An application

- The Water Jugs (Jackson and Willis, 1995) :

  `https://www.youtube.com/watch?v=6cAbgAaEOVE`

- More containers?

  `http://blancosilva.github.io/post/2016/07/29/decanting.html`

Paper examples: network

Trasport Costs and Economic Development: Storeygard (2016)

Storeygard, Adam (2016). "Farther on down the road: transport costs, trade and urban growth in sub-Saharan Africa" RES 123(1): 139-176.

Motivation

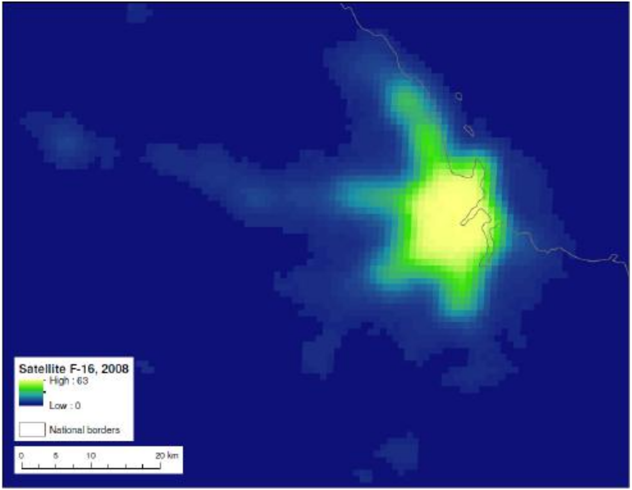
How inter-city connectivity determines the income of sub-Saharan African cities.

Contribution

Using road quality/type, lights, and oil price shock, estimates how growth is propagated far away from the city port. Negative elasticity of city economic activities wrt transport costs. Effect is heterogeneous in road quality (paved/unpaved).

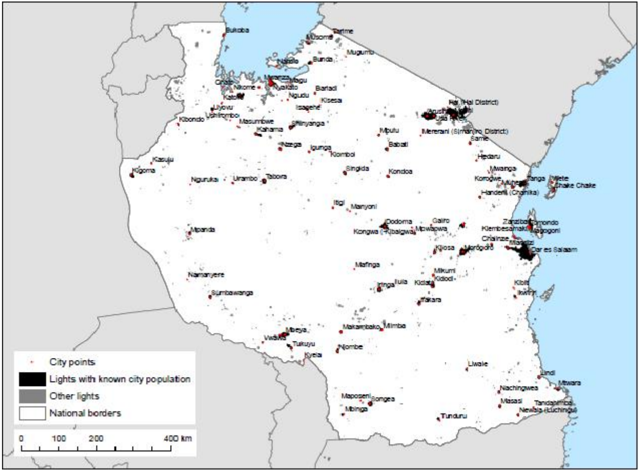
Paper examples: network

Luminosity (2008) in Dar el Salaam



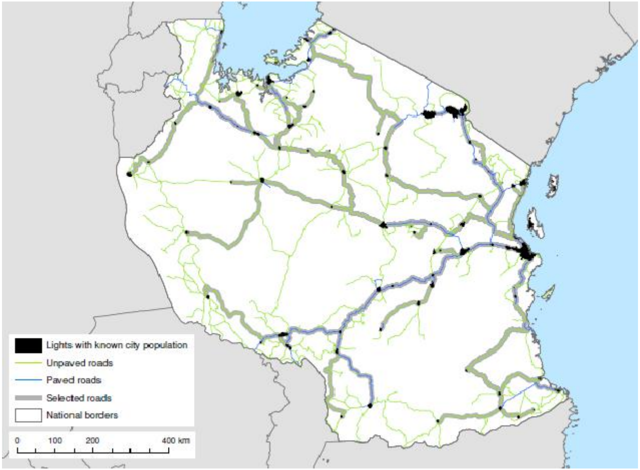
Paper examples: network

City Distribution in Tanzania



Paper examples: network

Roads connection to Dar el Salaam (Tanzania)



Donaldson, Dave and Hornbeck, Richard (2015). "Railroads and American Economic Growth: A "Market Access" Approach" QJE

Motivation

- Evaluate the role of railway construction for US economic growth.

Debate

- Local effect vs Aggregate effect for program evaluation. Particularly relevant if spillovers are present (SUTVA violation).

Market Access

Railroads and American Economic Growth: Donaldson and Hornbeck (2015)

Contribution

- Combine transportation network over time and census data to show that expansion of the railway network fostered both local and aggregated economic growth.
- Theory-based application of intra-country trade model (Eaton and Kortum, 2002) in a reduce form framework to quantify the spillover effect of infrastructure project.

Findings

- Railways expansion fostered US economic growth. Aggregate effects are considerably larger than local effects (due to higher "market access").

Market Access

Railroads and American Economic Growth: Donaldson and Hornbeck (2015)

Key expression that provides a first-order approximation to counties' market access:

$$MA_o \approx \sum_d \tau_{od}^{-\theta} N_d \tag{1}$$

where:

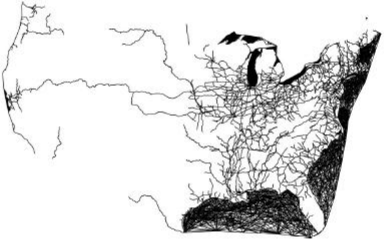
- MA_o: Market Access at origin (o)
- tau_{od}: bilateral transportation cost between origin (o) and destination (d)
- N_d: Population count at destination (d)

In our application, we are interested in deriving τ_{od}.

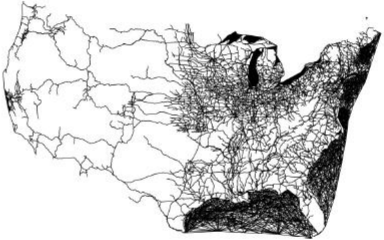
Market Access

Railway expansion 1870-1890

C. Natural Waterways, Canals, and 1870 Railroads

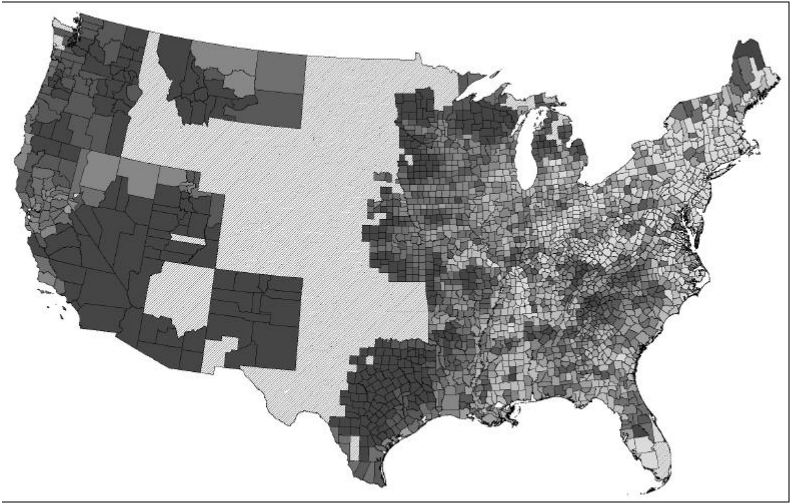


D. Natural Waterways, Canals, and 1890 Railroads



Market Access

Change in Market Access 1870-1890



Market Access

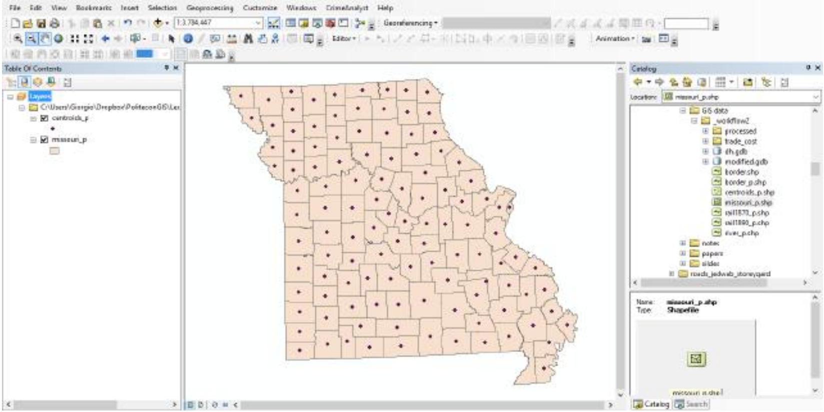
Replication DH network construction - Missouri

- We are going to construct the bilateral transportation costs in 1870 and 1890 for Missouri
- Prepare the network elements: net_prep.py
- Solve the best routes problem in both periods: trade_cost_od.py

Market Access

Replication DH network construction - Missouri

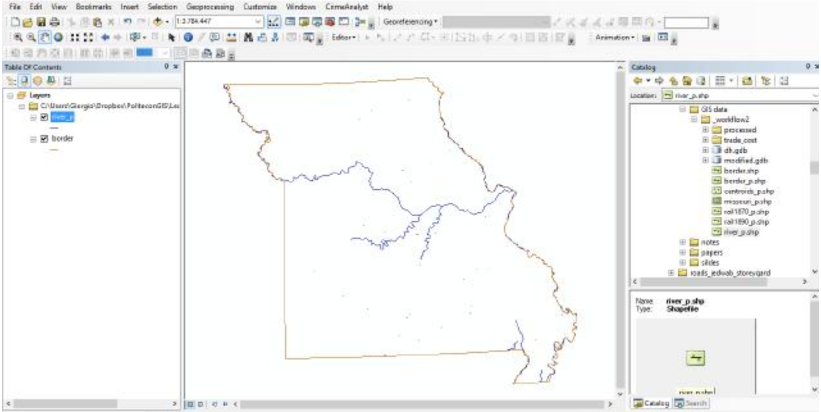
Centroids and Counties 1890



Market Access

Replication DH network construction - Missouri

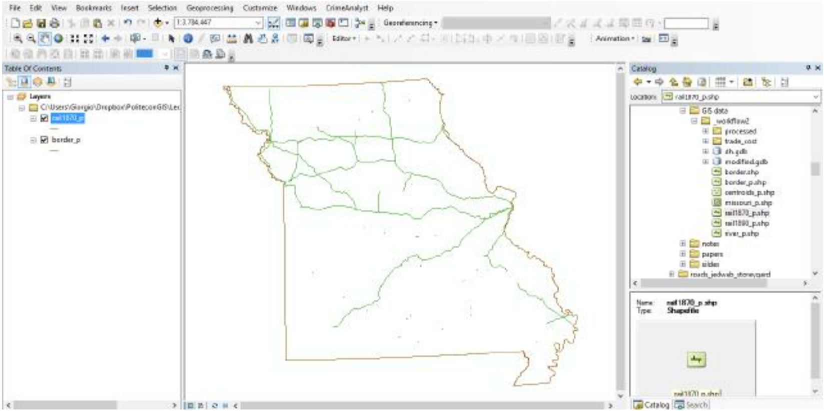
River Distribution



Market Access

Replication DH network construction - Missouri

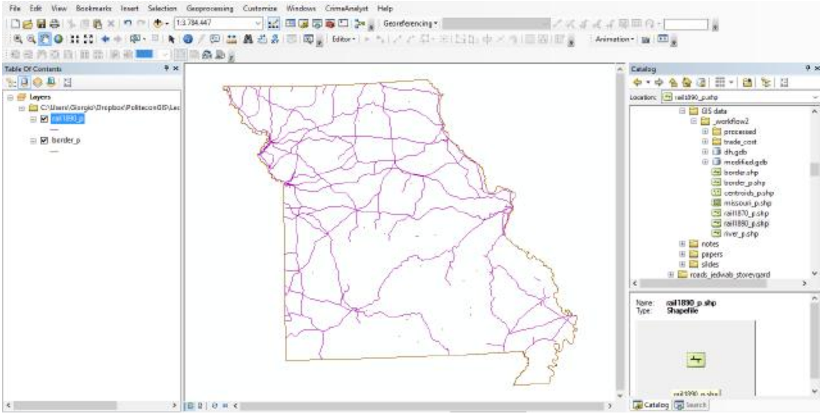
Railway expansion 1870



Market Access

Replication DH network construction - Missouri

Railway expansion 1890



Market Access

Replication DH network construction - Missouri

Cost Parameters (Fogel, 1962)

- River = 0.0049 USD (tons/mile)
- Rail = 0.0063 USD (tons/mile)
- Wagon Routes = 0.231 USD (tons/mile)

We are going to ignore Transshipment Cost (0.50 USD) in this replication.

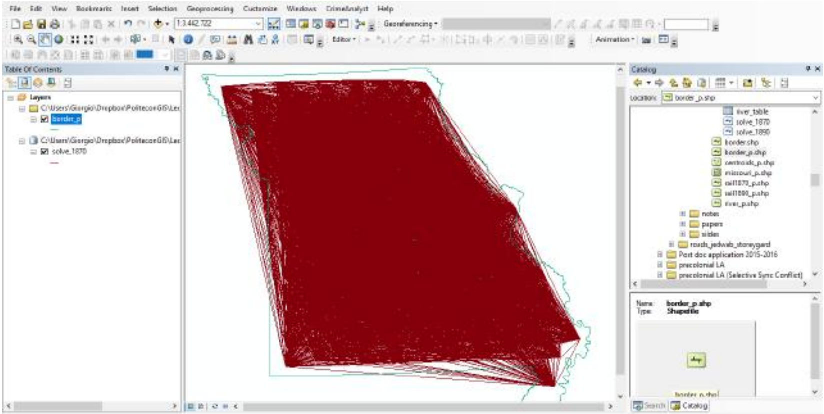
What we are going to do is to prevent switching transportation modes.

Total Cost = Parameter x Lenght (in miles)

Solving the Best Routes Problem

- We are going to solve the network in 1870 and 1890
- We obtain a matrix of 114x114 of bilateral transportation costs
- This is the τ in the Market Access expression

Solution Output in OD matrix

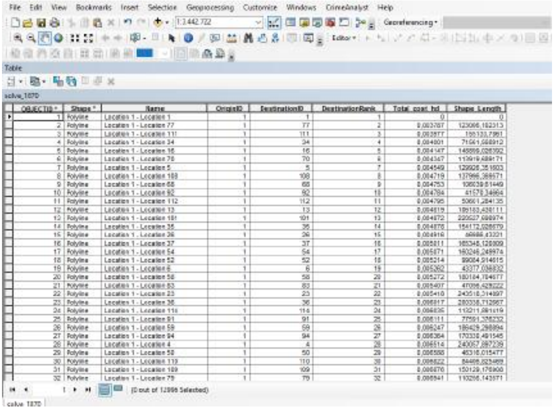


Use Closest Facility to visualize actual best routes solution.

Market Access

Replication DH network construction - Missouri

Solution Output Table in OD matrix



Table

solve_1870

OBJECTID *	Shape *	Name	OriginID	DestinationID	DestinationRank	Total_cost_hd	Shape_Length
1	Polyline	Location 1 - Location 1	1	1	1	0	0
2	Polyline	Location 1 - Location 77	1	77	2	0,003787	123086,182313
3	Polyline	Location 1 - Location 111	1	111	3	0,003977	155133,7961
4	Polyline	Location 1 - Location 34	1	34	4	0,004001	71561,568912
5	Polyline	Location 1 - Location 16	1	16	5	0,004147	148886,026392
6	Polyline	Location 1 - Location 70	1	70	6	0,004347	113919,689171
7	Polyline	Location 1 - Location 5	1	5	7	0,004549	129926,351603
8	Polyline	Location 1 - Location 108	1	108	8	0,004719	137966,366571
9	Polyline	Location 1 - Location 68	1	68	9	0,004753	108038,61449
10	Polyline	Location 1 - Location 92	1	92	10	0,004784	41578,34664
11	Polyline	Location 1 - Location 112	1	112	11	0,004795	50661,284135
12	Polyline	Location 1 - Location 13	1	13	12	0,004819	186183,438111
13	Polyline	Location 1 - Location 101	1	101	13	0,004872	220627,698974
14	Polyline	Location 1 - Location 35	1	35	14	0,004876	154172,028679
15	Polyline	Location 1 - Location 26	1	26	15	0,004916	69886,43321
16	Polyline	Location 1 - Location 37	1	37	16	0,005011	165348,128009
17	Polyline	Location 1 - Location 54	1	54	17	0,005071	160246,249974
18	Polyline	Location 1 - Location 52	1	52	18	0,005214	99084,914615
19	Polyline	Location 1 - Location 6	1	6	19	0,005262	43317,036832
20	Polyline	Location 1 - Location 58	1	58	20	0,005272	180184,784677
21	Polyline	Location 1 - Location 83	1	83	21	0,005407	47086,429222
22	Polyline	Location 1 - Location 23	1	23	22	0,005410	243518,314897
23	Polyline	Location 1 - Location 36	1	36	23	0,006017	280338,712867
24	Polyline	Location 1 - Location 114	1	114	24	0,006036	113211,881619
25	Polyline	Location 1 - Location 91	1	91	25	0,006111	77591,376232
26	Polyline	Location 1 - Location 59	1	59	26	0,006247	186429,298894
27	Polyline	Location 1 - Location 94	1	94	27	0,006364	170338,491546
28	Polyline	Location 1 - Location 4	1	4	28	0,006514	240067,897239
29	Polyline	Location 1 - Location 50	1	50	29	0,006588	46316,015477
30	Polyline	Location 1 - Location 110	1	110	30	0,006822	84466,825469
31	Polyline	Location 1 - Location 109	1	109	31	0,006876	150129,176906
32	Polyline	Location 1 - Location 79	1	79	32	0,006941	110266,143571

(0 out of 12996 Selected)

solve_1870