GIS for Economists 5

Giorgio Chiovelli Sebastian Hohmann Tanner Regan

21/06/2021

Overview

The plan for today

Spatial Network in Applied Economics

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

Spatial Network Analysis in GIS

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

Paper examples: network

Trasport Costs and Economic Development: Storeygard (2016)

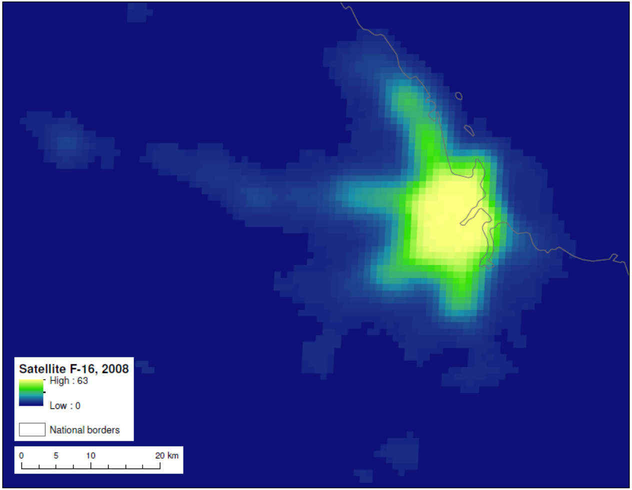
Storeygard, Adam (2016). "Farther on down the road: transport costs, trade and urban growth in sub-Saharan Africa" RES 123(1): 139-176.

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).

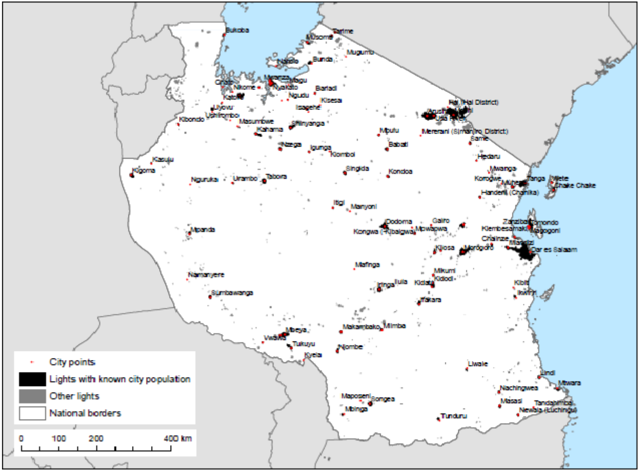
Paper examples: network

Luminosity (2008) in Dar el Salaam



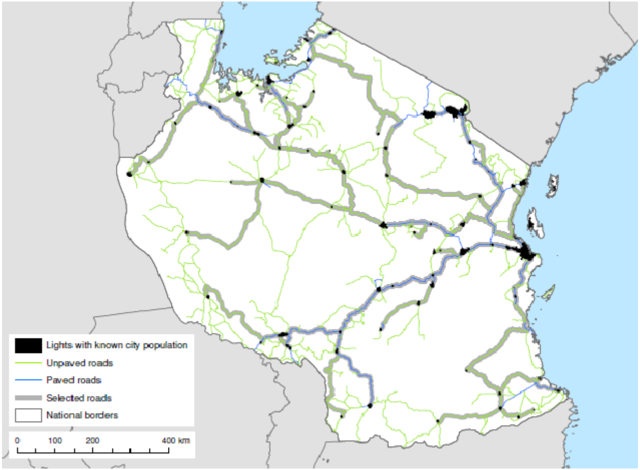
Paper examples: network

City Distribution in Tanzania



Paper examples: network

Roads connection to Dar el Salaam (Tanzania)



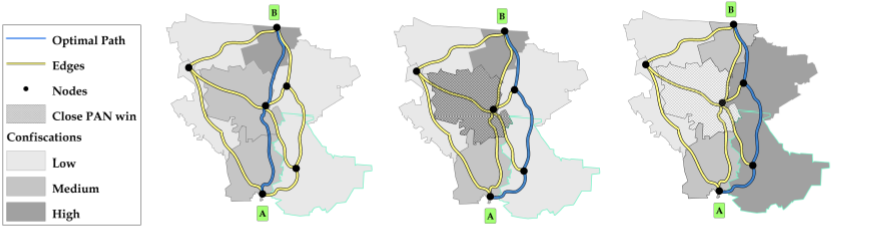
Dell, Melissa (2016). “Trafficking Networks and the Mexican Drug War” AER

Motivation
How crime repression policy affects spatial pattern of violence and crime .

Contribution
Using close elections, disaggregated crime data, and road network, to identify how crime reacts to crackdown policy. Quantification of externalities and unintended consequences of crime repression.

Paper examples: network

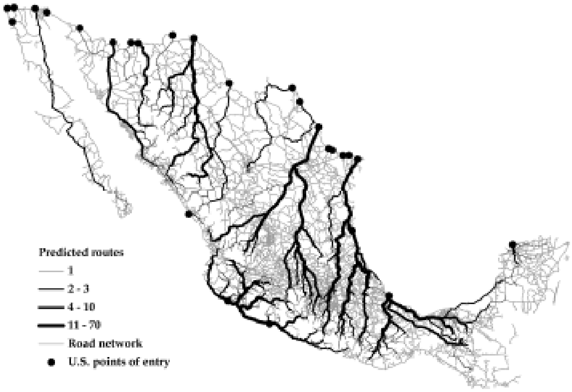
Spillover Methodology



(a) Legend (b) Basic environment (c) Close PAN victory increases costs; routes change (d) Do confiscation patterns also change?

Paper examples: network

Roads Network and Predicted Trafficking Routes



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)

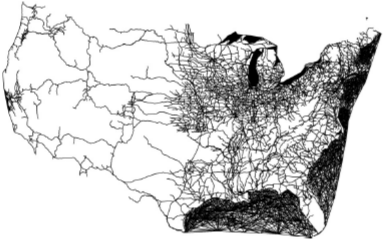
In our application, we are interested in deriving τ_{od}.

Market Access

Railway expansion 1870-1890

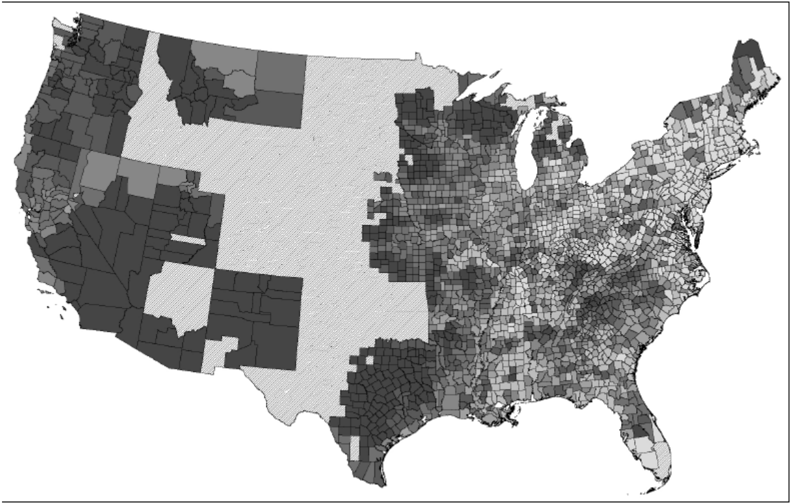
C. Natural Waterways, Canals, and 1870 Railroads

D. Natural Waterways, Canals, and 1890 Railroads



Market Access

Change in Market Access 1870-1890



Market Access

Regression Framework

$$lnV_{ot} = \beta ln(MA_{ot}) + \delta_o + \delta_{st} + f(x_o, y_o)\delta_t + \epsilon_{ot} \qquad (2)$$

- V_{ot} agricultural land values in county o and year t
- MA_{ot} market access
- δ_o county fixed effect
- δ_{st} state-by-year fixed effects
- $f(x_o, y_o)\delta_t$ cubic polynomial in county latitude and longitude interacted with year effects

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)**

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
  - Cost Parameters (Fogel, 1962)
    - River = 0.0049 USD (tons/mile)
    - **Rail = 0.0063 USD (tons/mile)**
    - **Wagon Routes = 0.231 USD (tons/mile)**

Total Cost = Parameter x Lenght (in miles)

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
- 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

- **Networkx**
  - Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
  - Much more than "just" spatial network (classic graphs, random graphs, and synthetic networks)
  - Convenient way of importing shapefile as network graph
- Install: conda install -c anaconda networkx
- Conflict with Python 3.9.
  - Solution: `https://github.com/smicallef/spiderfoot/issues/1124`

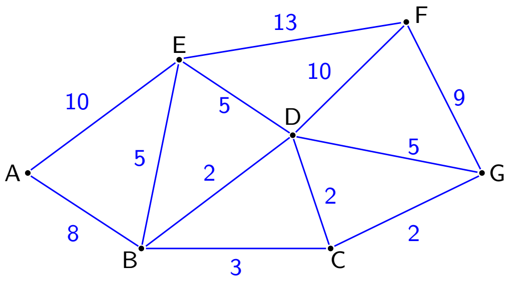
Workflow

- Prepare your network
  - Clean your line (roads, railways) shapefile
  - Set your cost parameter, if needed. e.g. Railways faster than roads
- Construct you graph
  - Check connectivity
  - Add attribute to nodes and edges
- Solve for the best route problem (Dijkstra's algorithm)

→ open `network.ipynb`

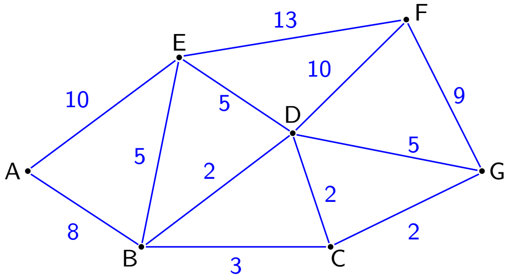
Dijkstra's algorithm

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



Dijkstra's algorithm

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

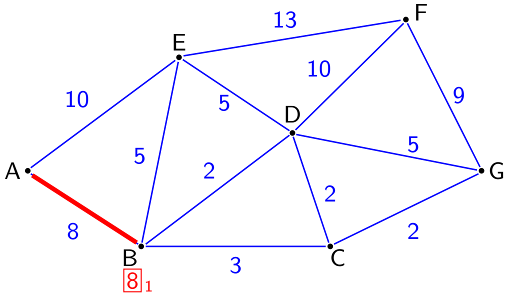


Algorithm

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**.

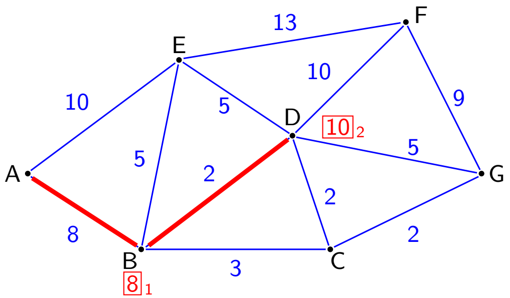


Algorithm

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**.

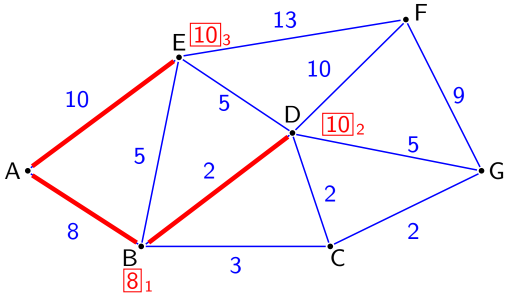


Algorithm

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 $= 0$)

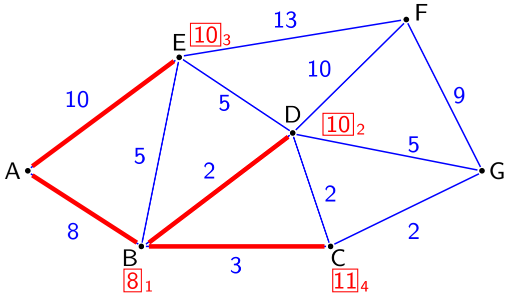
2 calculate distance through each unvisited neighbour

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**.

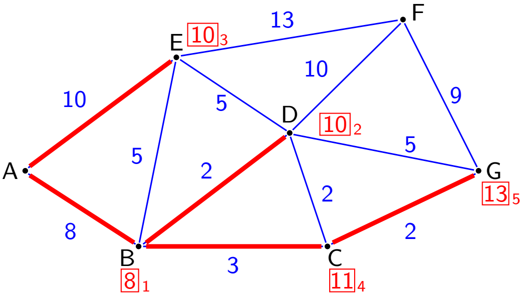


Algorithm

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**.

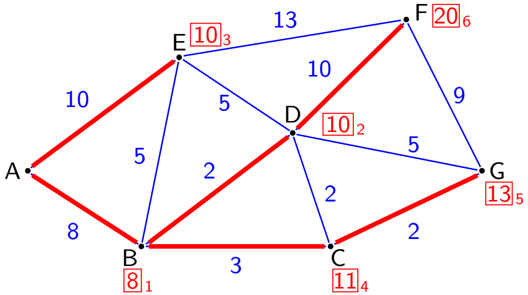


Algorithm

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://youtu.be/5ujdpliqP6Y`
- More containers?
  `http://blancosilva.github.io/post/2016/07/29/decanting.html`