Flexible Network Design Utilizing Non Strict Modeling Approaches

Fabion Kauker - fkauker@3-gis.com

INFORMS ALIO International 2019 Cancun

Background

- Challenges modelling the exact modelling requirements of domain problems
- Scalability not guaranteed
- Not many UX oriented optimization software solutions
- Interactive software lacks intuition when dealing with “solutions”
- Many iterations needed to add user feedback
- Models are good at one specific application sometimes for only one customer
- Hard to pick the best modelling option with limited information
- Feedback on solutions can break the modelling approach

Geographic Information Systems (GIS)

Many options for software
- ESRI
- MapInfo
- QGIS

Key concepts

Geometry
- Points
- LineString
- Polygon

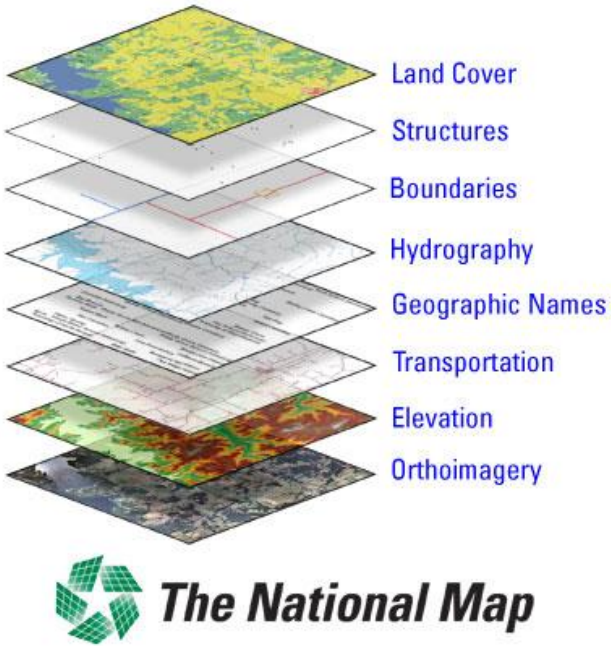
Attributes

Formats

SHP, TAB, GeoJSON

```
{"type": "FeatureCollection",
 "features": [{
  "type": "Feature",
   "geometry": {
       "type": "Point",
       "coordinates": [125.6, 10.1]
   },
   "properties": {
       "name": "Dinagat Islands"
   }
}]}
```

Open Data:
https://openaddresses.io/
https://www.openstreetmap.org/



Source: https://cegis.usgs.gov/images/national_map_layers.jpg

GeoJSON rendering on Github

Source: https://github.com/fhk/test_data

Graphs

Given that we have some geometry we want to derive the graph structure.

Where:
vertex/node == Point or LineString endpoint
and
edge/arc == LineString endpoint to LineString endpoint

Directional vs. Undirected - degree 2 nodes = degree 1 nodes

Why Homebaked v. Open Source?

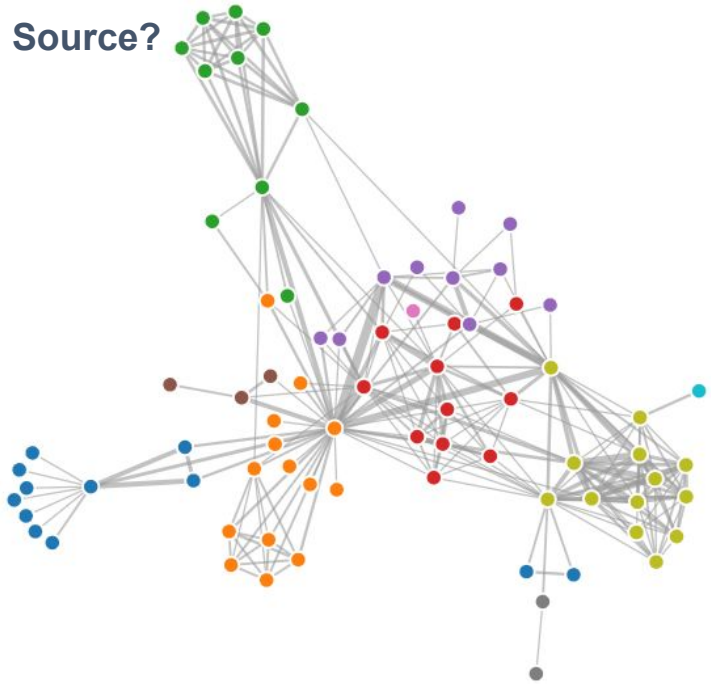
Python
networkx
graph-tool

Baked in algos
Edge and node access

Connected components

Used for path computation

Representations:
adjacency list or matrix



Source: https://observablehq.com/@d3/force-directed-graph

MIP Formulations

(1) Assignment MIP formulation

$$(1.1)\ min \sum_{ij} p_{ij}.x_{ij}$$

s.t.

$$(1.2)\ \sum_{j} x_{ij} = 1$$

$$(1.3)\ \sum_{j} x_{ij} - c_i.z_i \leq 0$$

$$(1.4)\ \sum_{i} z_i = 1$$

(2) Network flow formulation

$$(2.1)\ min \sum_{i} c_i.z_i + \sum_{ij} p_{ij}.x_{ij}$$

s.t.

$$(2.2)\ \sum_{ij} x_{ij} - \sum_{ki} x_{ki} = d_i$$

$$(2.3)\ \sum_{i} x_{ij} \leq z_i * p_i$$

$$(2.4)\ x_{ij} - w.x_{ji} \leq 0$$

Algorithms/Heuristics

Prize Collecting Steiner Tree (PCST)
Find a sub-tree within G (the input graph) such that all points specified are connected.
"We call this variant the prize-collecting Steiner forest (PCSF) problem and adapt the algorithm of(Goemans & Williamson, 1995) for this variant." [1]
Algorithm has two stages.[2]

1. Growing stage, create clusters
2. Pruning stage, remove edges

Python & C++ implementation[3]

Teitz-Bart solution to p-median

1. Create potential assignments
2. Apply "search" to find new assignments outside of current set by removing then adding node
3. Continue till "cost" cannot be improved

Why trees you ask?
"While the p-median problem is NP-hard on a general graph, the problem can be solved in polynomial time on a tree" [6,7]

1) http://people.csail.mit.edu/ludwigs/papers/icml15_graphsparsity.pdf
2) http://people.csail.mit.edu/ludwigs/papers/dimacs14_fastpcst.pdf
3) https://github.com/fraenkel-lab/pcst_fast
4) https://cran.r-project.org/web/packages/tbart/index.html
5) https://pubsonline.informs.org/doi/abs/10.1287/opre.16.5.955
6) Goldman AJ (1971) Optimal center location in simple networks. Transp Sci 5:212–221
7) https://pdfs.semanticscholar.org/2094/882d425fe9b3f668eaafbcf8ac0bba478b5f.pdf

Domain problems

Shipping

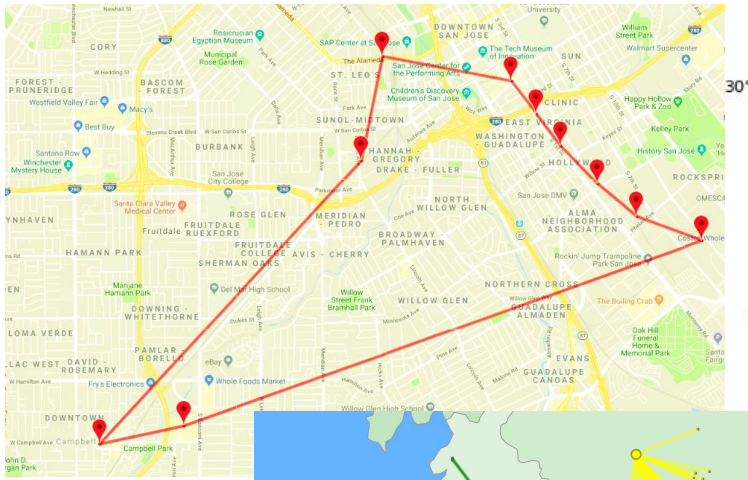
Facility location

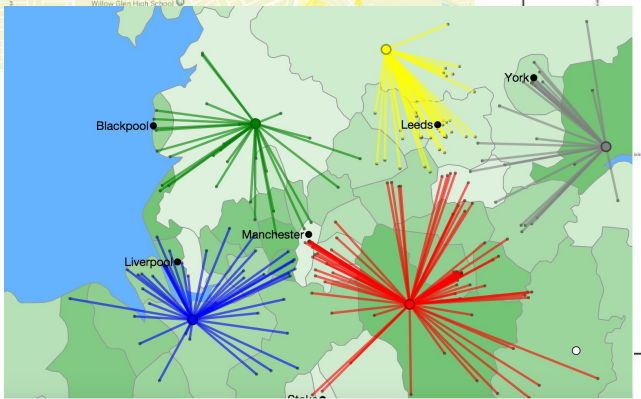
Infrastructure design

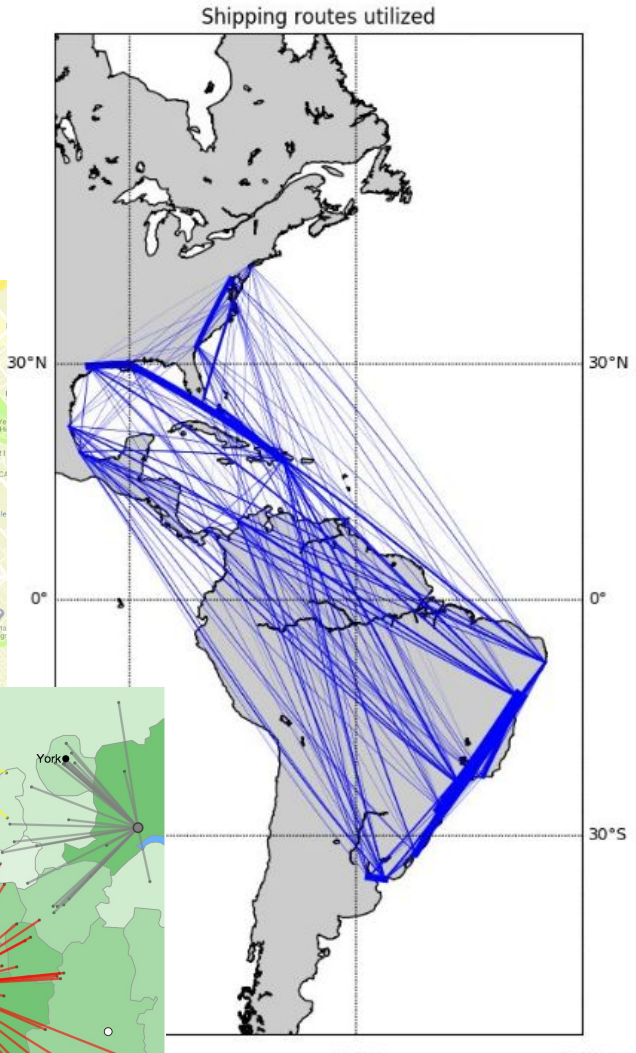
Logistics

Scheduling

...







Source: http://examples.gurobi.com/facility-location/

Modelling Variations

- Non overlapping path assignments
- Hub capacity
- Edge reuse on path or flow models
- Node and/or edge capacity
- Path length & behaviours
- Multi-tiered flow + path
- Time period staged

Time v. Optimality v. solution space v. complexity

Size of model

Non-linear scaling models
- Street centreline to dual sides

Optimality checking

Many alternate solutions
- Run time
- Reproducibility

Testing

10’s of nodes and edges

~40GB OSM data
~12GB Open Address data
0.0012257362589129MB per KM in The US

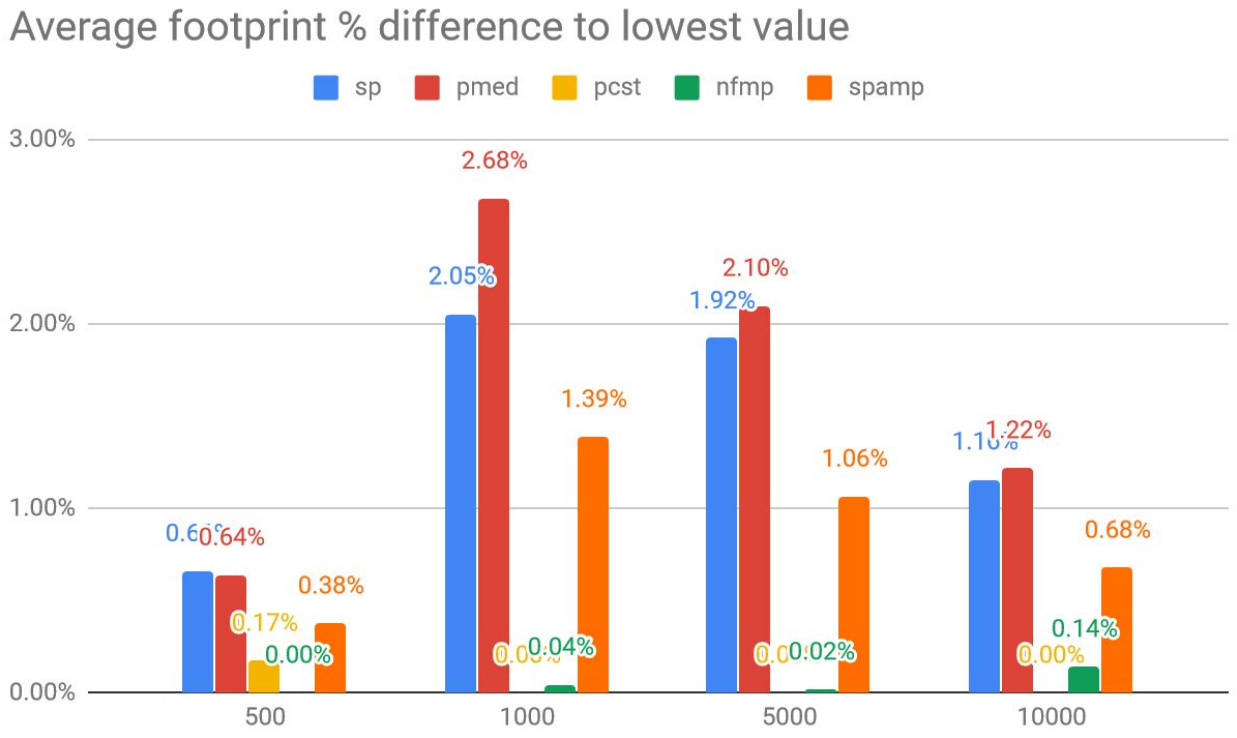
Toy problem

Goldilocks zone for modelling

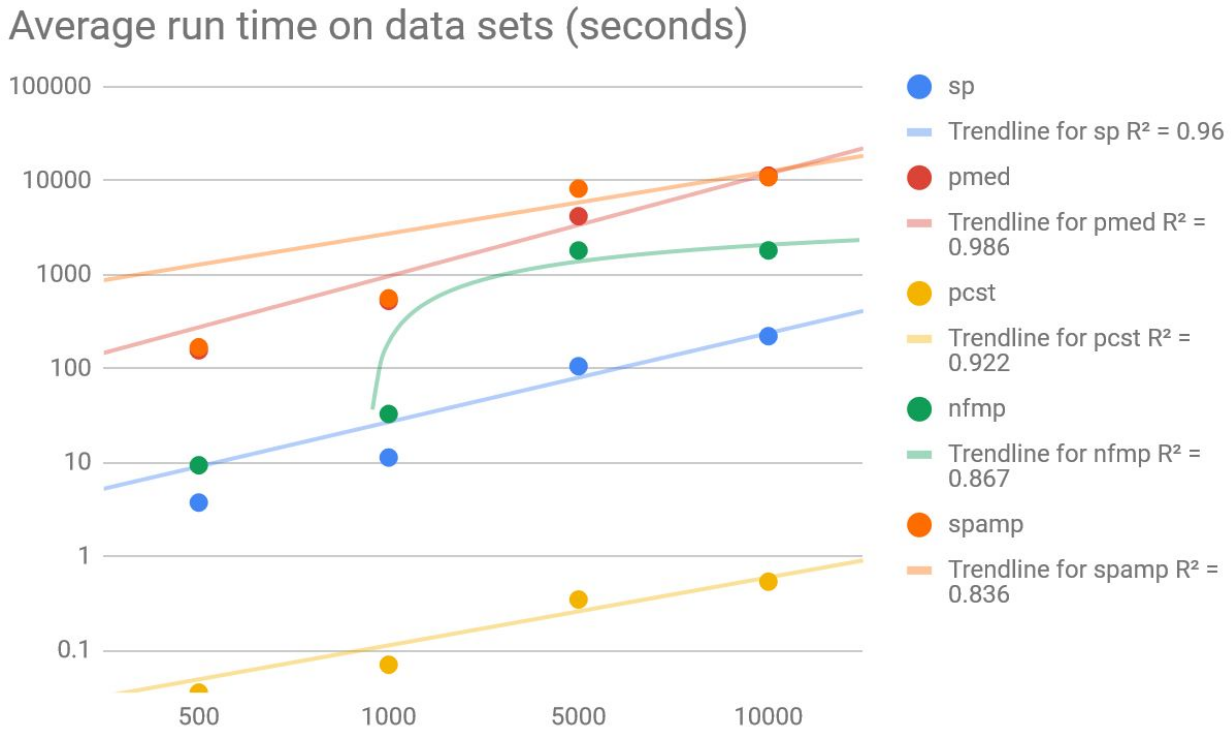
World data

img: https://www.pinterest.co.uk/pin/569916527817569073/

Benchmarks


Average footprint % difference to lowest value
sp
pmed
pcst
nfmp
spamp
3.00%
2.00%
1.00%
0.00%
2.68%
2.05%
2.10%
1.92%
1.39%
1.06%
1.22%
0.64%
0.38%
0.17%
0.00%
0.04%
0.02%
0.14%
0.00%
0.68%
500
1000
5000
10000

Benchmarks


Average run time on data sets (seconds)
100000
10000
1000
100
10
1
0.1
500
1000
5000
10000
sp
Trendline for sp R² = 0.96
pmed
Trendline for pmed R² = 0.986
pcst
Trendline for pcst R² = 0.922
nfmp
Trendline for nfmp R² = 0.867
spamp
Trendline for spamp R² = 0.836

Example

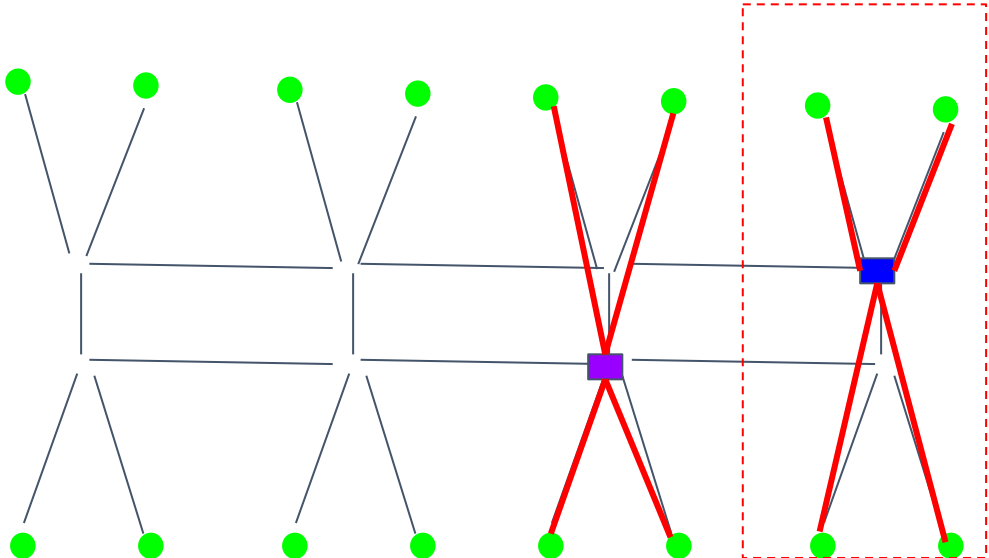
- We want to create “clusters” of at most size n
- Traditionally these are drawn manually in a GIS
- We could fully automate this with a MIP
- Leads to a 80-90% solution
- 80:20 rule…
- Is there a better way
- Let’s use a workflow and *pcst-fast*

PCST

1. Draw boundary around each group
2. Find solution
3. Append to working data
4. Validate solution

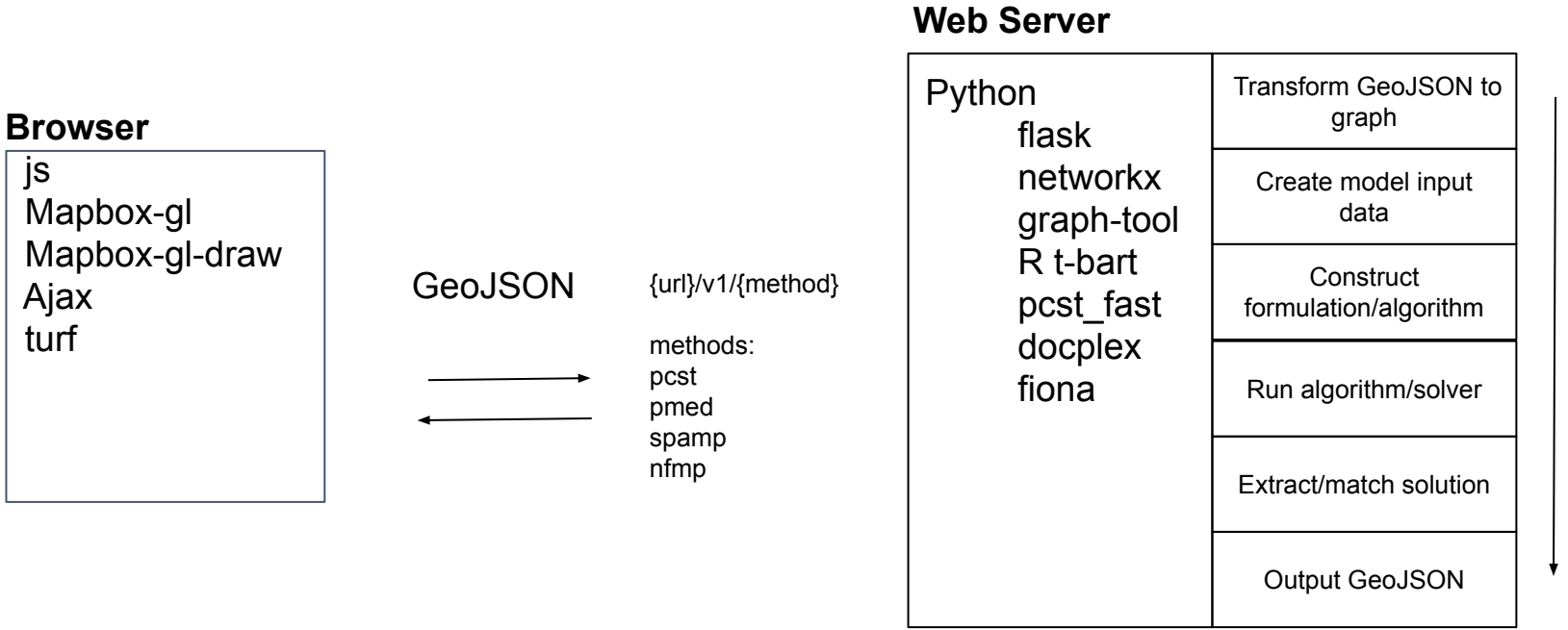
PAMP

1. Run solve
2. Validate solution
3. Edit solution or input
4. Rerun
5. Repeat



Demo

How?


Browser
js
Mapbox-gl
Mapbox-gl-draw
Ajax
turf
GeoJSON
{url}/v1/{method}
methods:
pcst
pmed
spamp
nfmp
Web Server
Python
flask
networkx
graph-tool
R t-bart
pcst_fast
docplex
fiona
Transform GeoJSON to graph
Create model input data
Construct formulation/algorithm
Run algorithm/solver
Extract/match solution
Output GeoJSON

Questions?

Github - https://github.com/fhk/link_src & https://github.com/fhk/test_data

Contribute models

Further learning
- Solver tuning
- Huertic implementation
- Tiers
- RL approaches

Publication to PhD

Come solve industry problems - roles at 3-GIS

Contact:
fkauker@3-gis.com