Louis Berger

wsp

Rapid Visuals of Regional Traffic Flows with Open-Source Tools

Presented by: Gabor Debreczeni

June 3, 2019

Presented to:

TRB NATIONAL TRANSPORTATION PLANNING APPLICATIONS CONFERENCE

Our Company and Team

Louis Berger | *Solutions for a better world*

wsp

- The work discussed here is by the Economic and Financial Analysis Team.
- We offer a wide range of capabilities in forecasting and economic analysis, advising government entities, private sponsors, equity investors, and lenders on toll highway projects, transit, infrastructure prioritization, and construction cost risk.

Full Travel Modeling

Caliper

TransCAD

Transportation Planning Software

- Great tools, but can take months to set up well and model runs can take days.
- For some projects, we have to be much quicker, so we came up with what we call the “rapid-routing approach”.

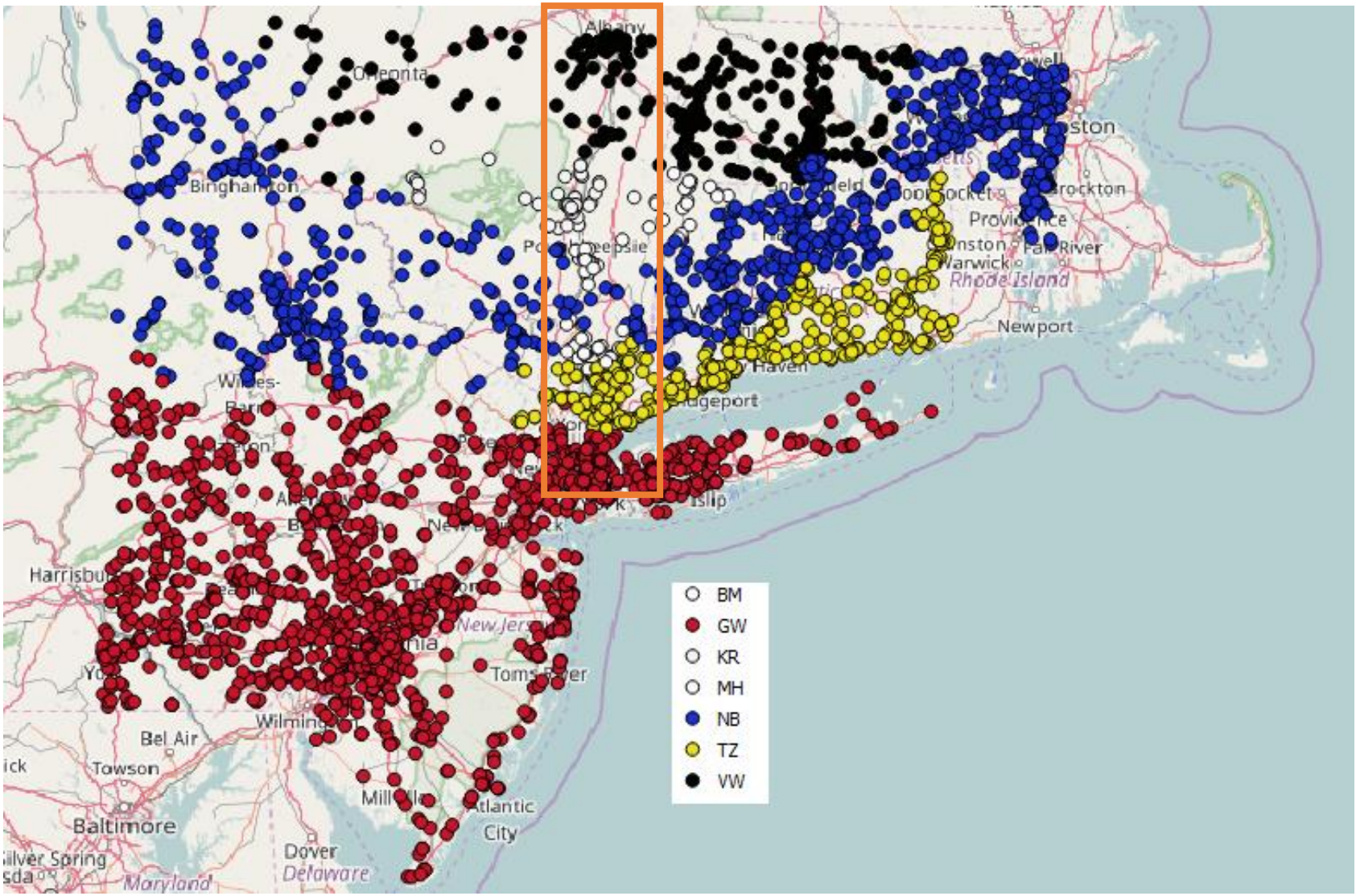
Tools Used in Rapid-Routing Approach

- We can visualize the behaviour of a regional road network in a few hours.
- Sketch-level approach but leads to useful insights.

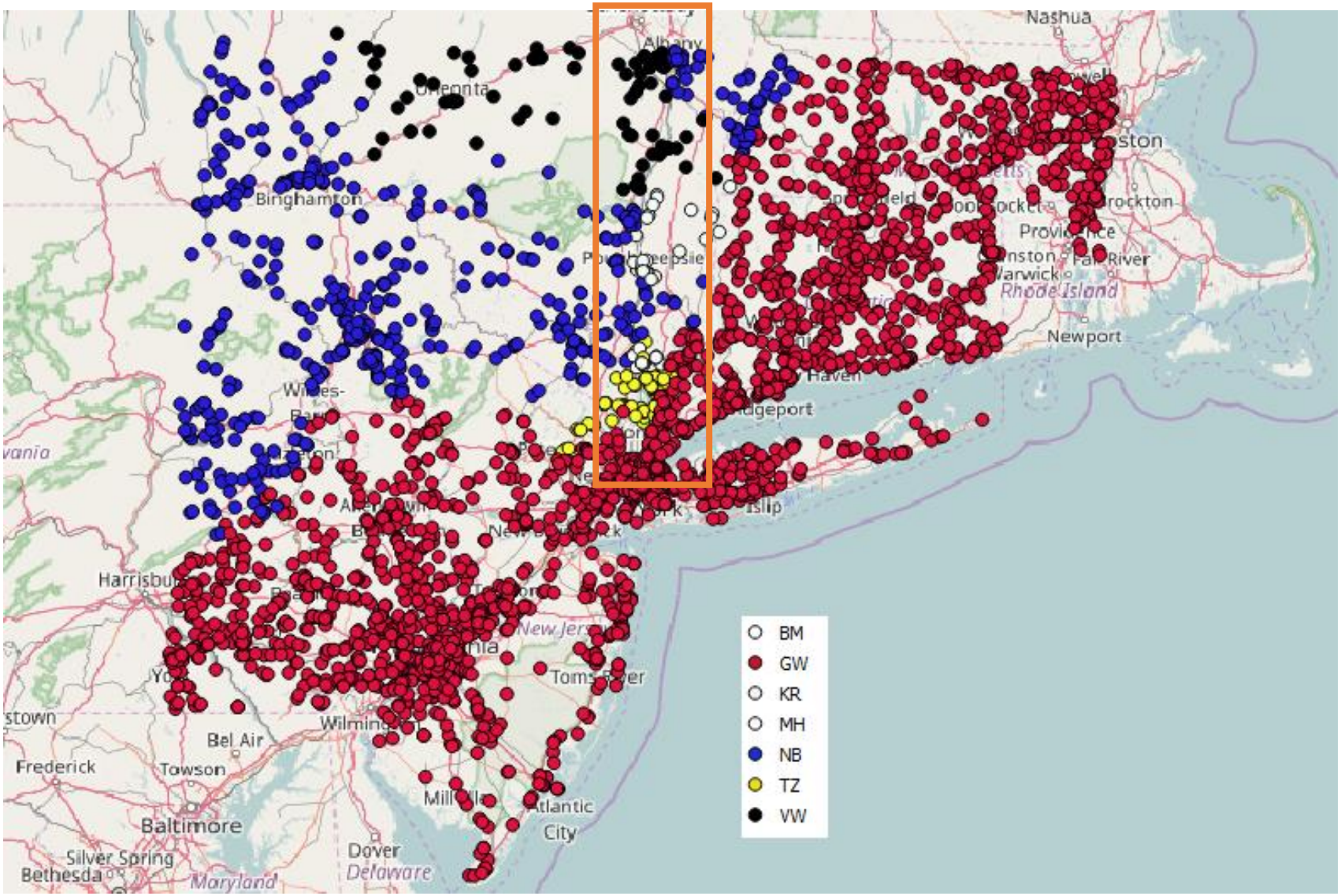
What does pgRouting do?

- The approach uses the pgRouting package's ability to:
  - Quickly calculate the fastest route between two points on a travel network
  - To report all network links used in the fastest route
  - As well as the travel time
- With the appropriate set-up, and once aggregated over a sufficiently large number of origins and destinations, this approach can reveal a great deal about the macro-level operation of a road network.

What is the Closest Bridge?



What Bridge is Used by a Plurality of Trips?



Drawbacks

- The pgRouting function requires a travel speed as an input, and as such it cannot be used to conduct a recursive equilibrium assignment process
- In the examples above, we assumed no-traffic conditions.
- This is why the Tappan Zee Bridge, frequently used as an alternative to the congested George Washington Bridge, has such little prominence in Figure 2. We can conclude that the Tappan Zee Bridge has little natural demand, and that it is primarily used by drivers seeking to avoid congestion elsewhere.

Captive Markets

- Blue: most captive – roads of interest used to all major cities.
- Green: least captive.
- A complete run of this kind takes approximately 10 to 30 minutes. Easy to fix mistakes and to iterate and optimize the outputs.

Final Comments

- A longer paper with example code is available at gabordebreczeni.com
- Feel free to get in touch at gabor.debreczeni@wsp.com

SQL Code Example

The example below uses Dijkstra's algorithm to find the fastest route between two links – referenced as "node" and "11044" (which, in our network, is a link in the center of a major city) in the SQL call shown in Figure 4. The "cost_toll" references the cost, in both time and money, of using each link in the network. For each link in the network, this call sets the column "AG53" equal to true if the fastest route uses the PR-53 highway, and false if it does not. Further examples of the use of the pgr_dijskra() function can be found at Web-7.

```
update pr_catchment_vot set AG53 = (
select exists (select 1 from
pgr_dijkstra(
    'SELECT id, source, target, cost_toll as cost FROM pr_roads_segmented',
    node, 11044, false
  ) as di join pr_roads_segmented on di.edge = pr_roads_segmented.id
where ref = 'PR-53' limit 1))
```

Figure 4: SQL Code to Find Whether Fastest Route Uses a Certain Highway

SQL Code Example

```
C:\Users\gdebreczeni\Documents\Matsim\Reverse engineering\SR-91>.\osmfilter bigm
ap.osm --out-count=highway
      4379     residential
      1985     service
      1878     turning_circle
       991     primary
       852     secondary
       691     tertiary
       335     footway
       243     motorway_link
       232     track
       192     motorway
       159     traffic_signals
        76     path
        57     crossing
        49     cycleway
        45     motorway_junction
        29     stop
        20     primary_link
        15     bus_stop
        14     secondary_link
        14     trunk
        14     unclassified
        11     turning_loop
         6     bridleway
         6     construction
         5     steps
         3     tertiary_link
         2     living_street
         1     street_lamp
```

Figure 5: Using Osmfilter to Find Road Types in Network

```
C:\Users\gdebreczeni\Documents\Matsim\Reverse engineering\SR-91>.\osmfilter bigm
ap.osm --keep="highway=motorway =motorway_link =motorway_junction =service =turn
ing_circle =primary =secondary =tertiary =crossing =stop =primary_link =secondar
y_link =trunk =unclassified =turning_loop =tertiary_link" > maptrim2.osm
```

Figure 6: Using Osmfilter to Simplify Network to Major Roads