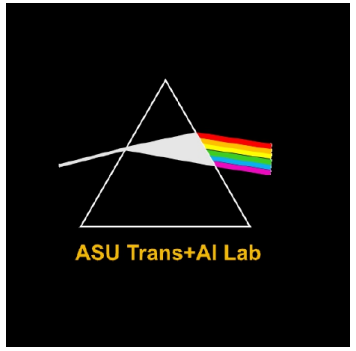
Transportation Modeling using GMNS standard, Open-Source AMS Tools

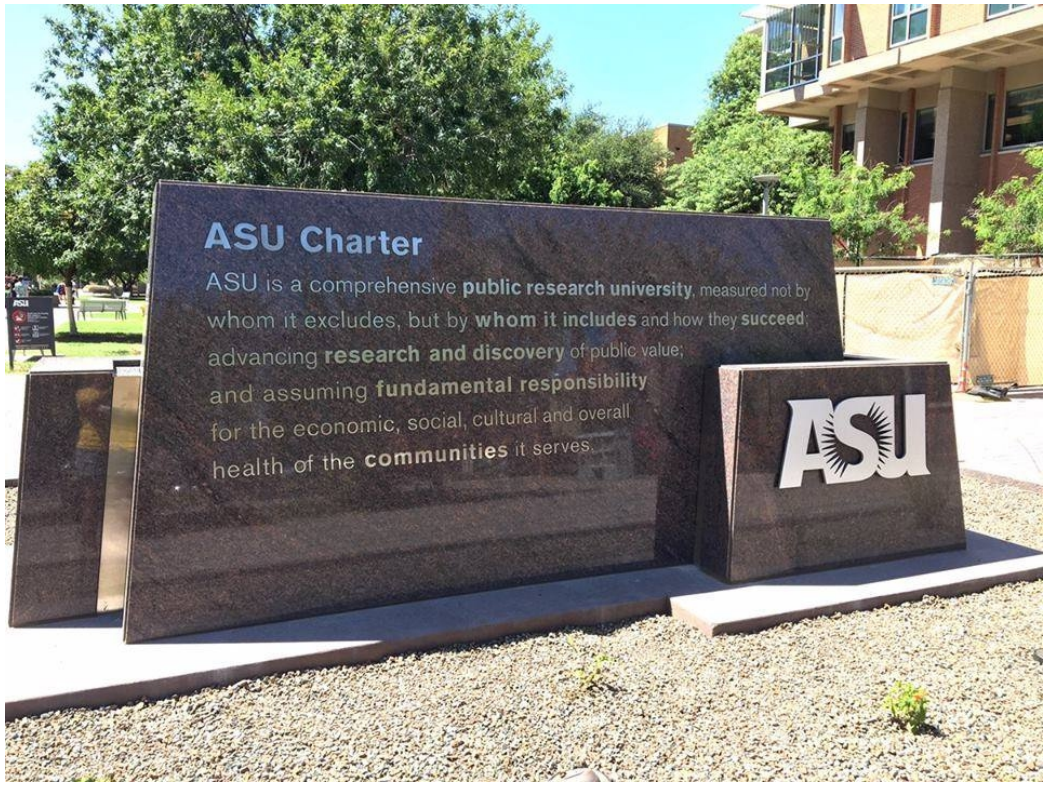
Developed through an early FHWA project: Effective Integration of Analysis, Modeling, and Simulation Tools
AMS DATA HUB CONCEPT OF OPERATIONS:
https://www.fhwa.dot.gov/publications/research/operations/13036/004.cfm

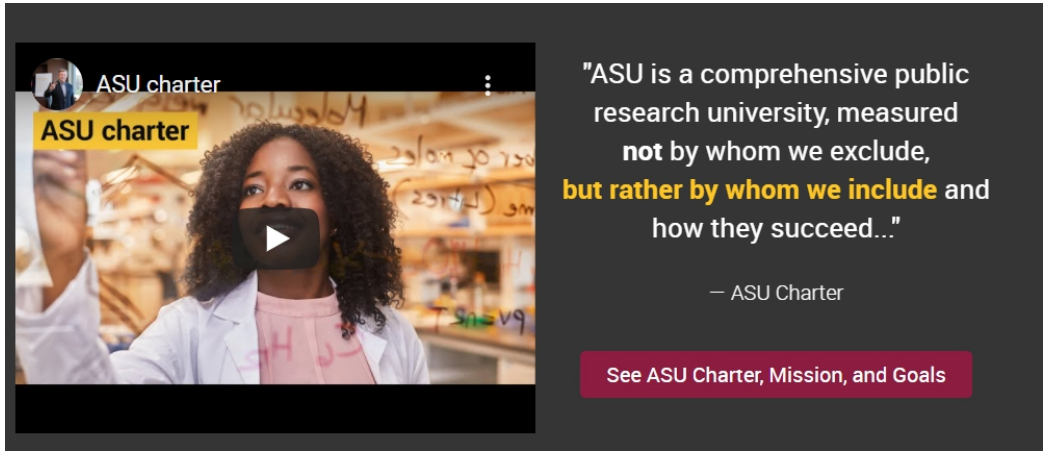
General Modeling Network Specification (GMNS): https://github.com/zephyr-data-specs/GMNS

Xuesong (Simon) Zhou (xzhou74@asu.edu)
https://github.com/asu-trans-ai-lab
https://asu-trans-ai-lab.github.io/website_openlayer_4GMNS/
School of Sustainable Engineering and the Built Environment
Arizona State University



Inclusive Excellence





https://inclusion.asu.edu/

Outline

1. Motivation
2. Integrated Modeling
3. Datahub
4. Connections to transportation Analaysis, Modeling and Simulation (AMS)
5. Large scale applications

1. Motivations

Adopting open network standard of GMNS

General Travel Network Format Specification is a product of Zephyr Foundation, which aims to advance the field through flexible and efficient support, education, guidance, encouragement, and incubation. Further details can be found in

https://zephyrtransport.org/projects/2-network-standard-and-tools/

Integrated graphic user interface and analysis package

(1) Provide an open-source code base to enable transportation researchers and software developers to expand its range of capabilities to various traffic management application.

(2) Present results to other users by visualizing time-varying traffic flow dynamics and traveler route choice behavior in an integrated environment.

(3) Provide a free, educational tool for students to understand the complex decision-making process in transportation planning and optimization processes.

GMNS

Process and Governance

- Overseen by a volunteer project management group (PMG), wh has met approximately quarterly over the past year
- Completed deliverables
  - Requirements
  - An initial specification
- Project Management Group

Joe Castiglione	SFCTA (chair)	Jeff Frkonja	Portland Metro
Michael Mahut	INRO	Scott Smith	Volpe / US DOT
Wu Sun	SANDAG	Natalia Ruiz Juri	University of Texas
Guy Rousseau	ARC	Song Gao	UMass Amherst
Chetan Joshi	PTV		

USDOT Volpe Center team
Scott.Smith@dot.gov
Ian.Berg@dot.gov

Timeline

- 2017: Zephyr: idea won the project "shark tank" at the TRB Planning Applications Conference
- 2018: FHWA: initial development and stakeholder outreach
- 2019: Zephyr and FHWA efforts came together
  - PMG established
  - Requirements formulated
  - Continuing development
- 2020: Initial public release
  - Initial release in January, with ongoing development throughout the year
    - Focus on traffic controls
  - Initial development of NEXTA modules to use GMNS
    - https://github.com/asu-trans-ai-lab
  - Conversion tools from OpenStreetMap

1. Motivations

- Existing technical barriers: (based on DTA user survey, TRB network modeling committee, 2009)
  - Require too many input data: 47%
  - Take too long to run: 35%
  - Model is unclear: 35%

Our goals

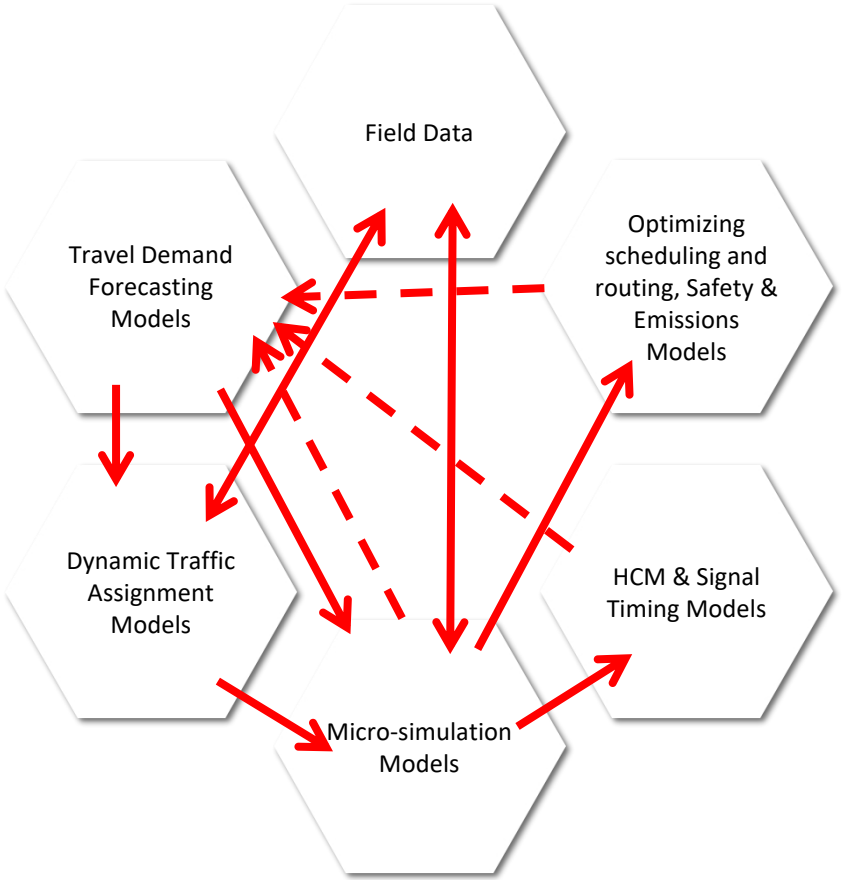
- Simplified data input from static traffic assignment
- Use parallel computing capability, simplified routing and simulation
- Open-source

  Visualization: Seeing is believing
  Excel Tools: Start from basics

Our Vision

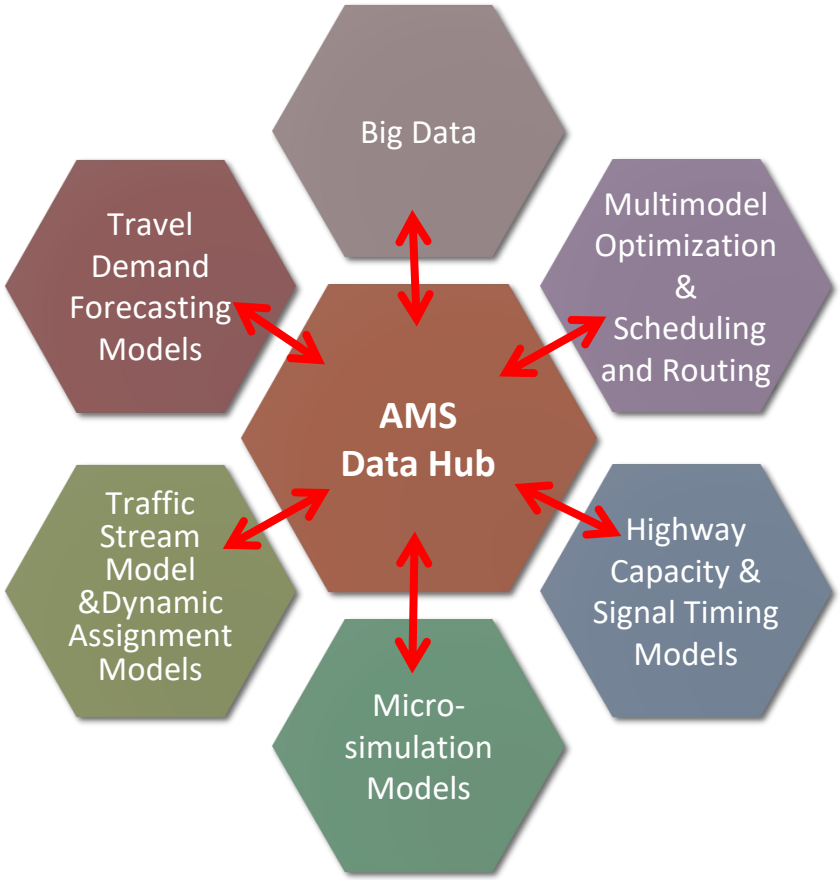
Current Practice

Ad Hoc

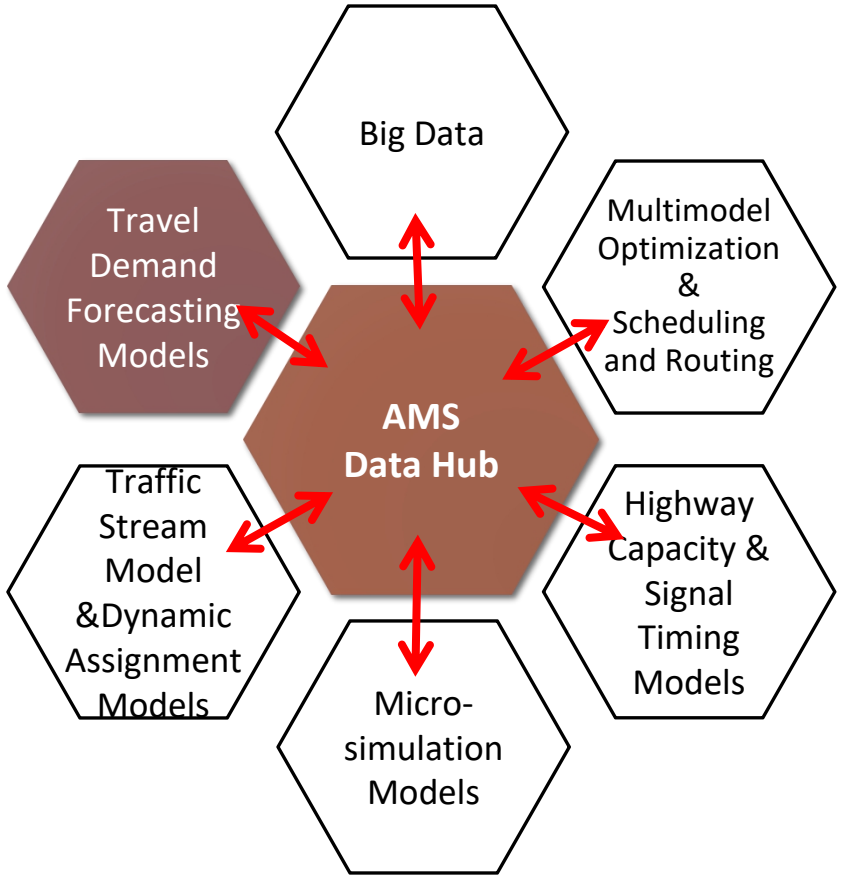


With Analysis, Modeling and Simulation (AMS) Data Hub

Systematic

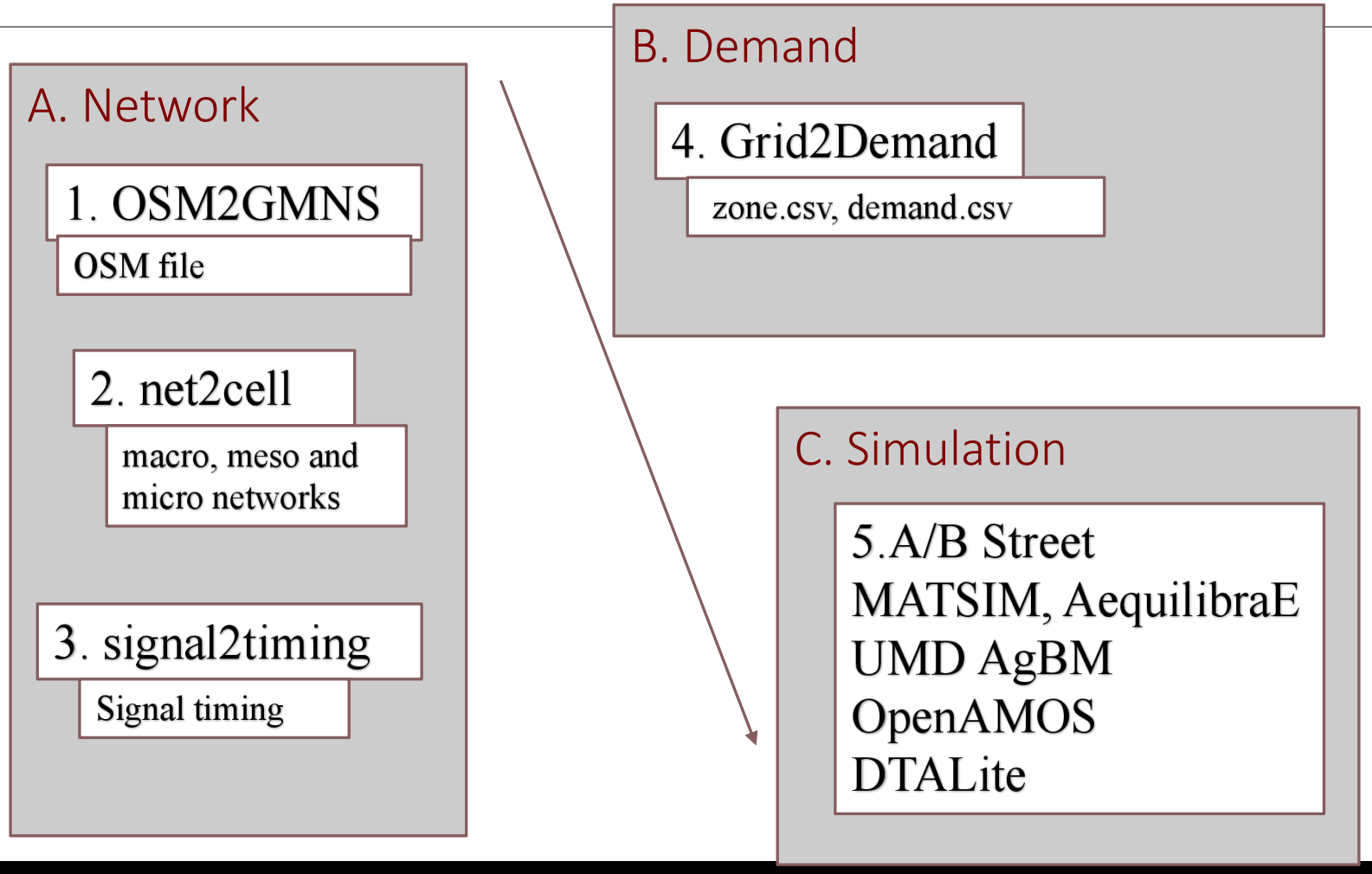


Data hub 0: Network



- Based on GMNS data format
  - node, link, movement
  - https://github.com/zephyr-data-specs/GMNS
- Regional planning models through GIS shape files
  - TransCAD
  - CUBE
  - VISUM
- Export network to GIS
- Subarea cut utility

Building 4-step Model For Your Campus Using Open-source Tool And GMNS



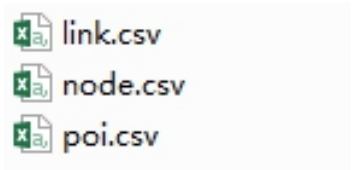
0.1 Open-Source Network Downloader

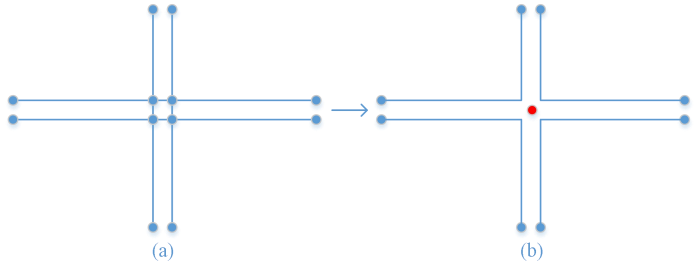
OSM2GMNS

Features

- 1) Easy to use, two lines of python code
- 2) Multimodal support (drivable, walkable, bikeable roads; railway; aeroway)
- 3) Network files in GMNS format
- 4) Transportation network modelling (intersection consolidation; point of interest)

```
import osm2gmns as og
net = og.getNetFromOSMFile('asu.osm')
og.outputNetToCSV(net)
```

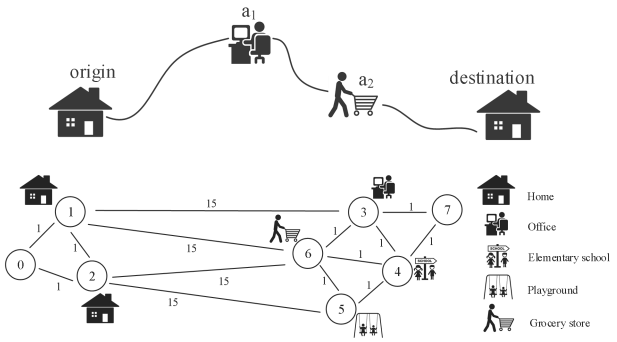


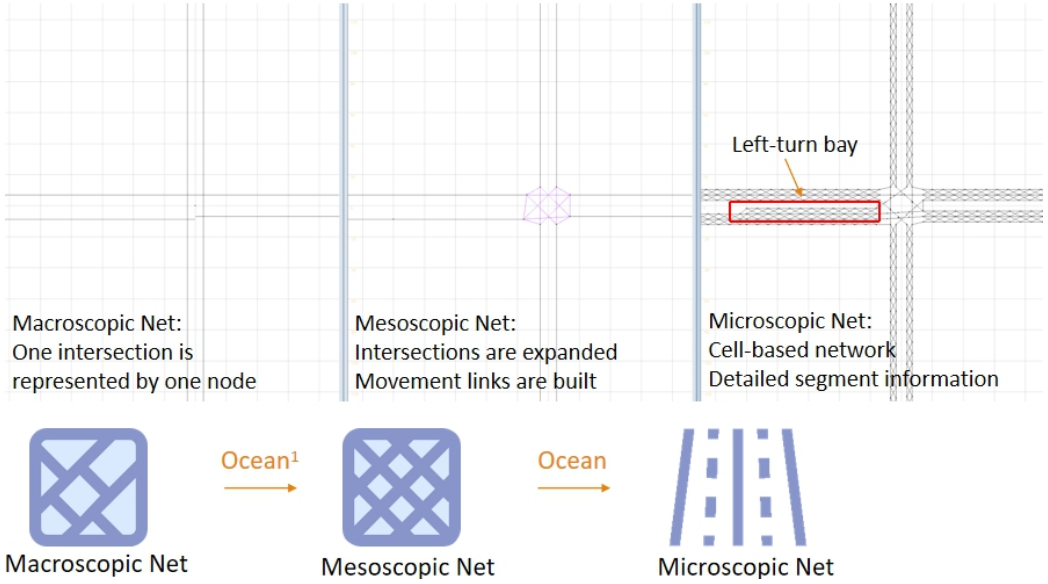


0.2 Multi-Level Network Convertor

https://pypi.org/project/net2cell/

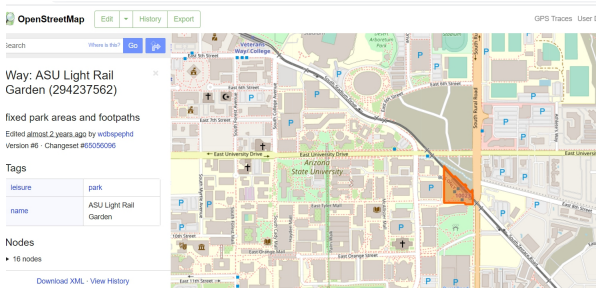
Open-source tool for creating cell-based (microscopic) and mesoscopic networks



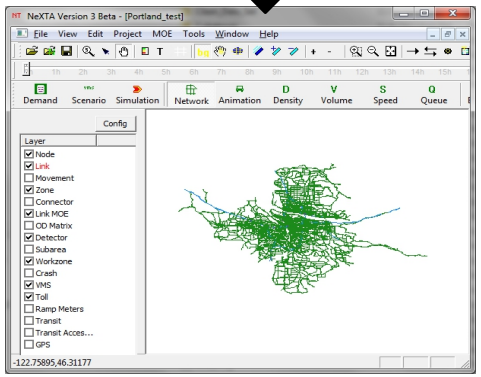


Openstreet Map Data Importing

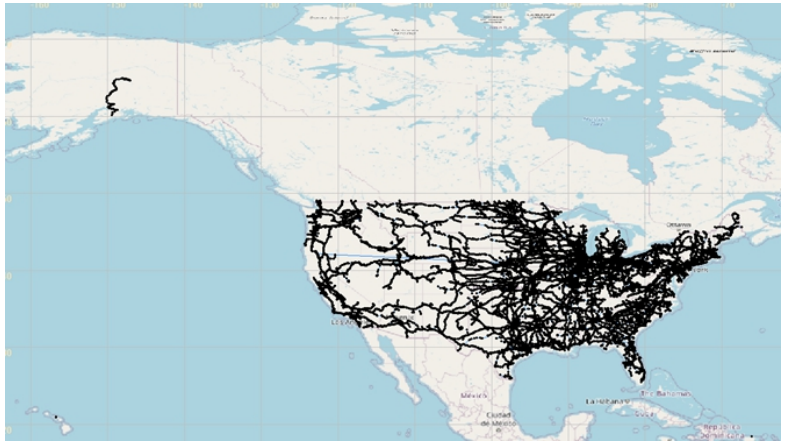
- Import node/link/zone shape to GMNS
  https://pypi.org/project/osm2gmns/



Openstreet Map

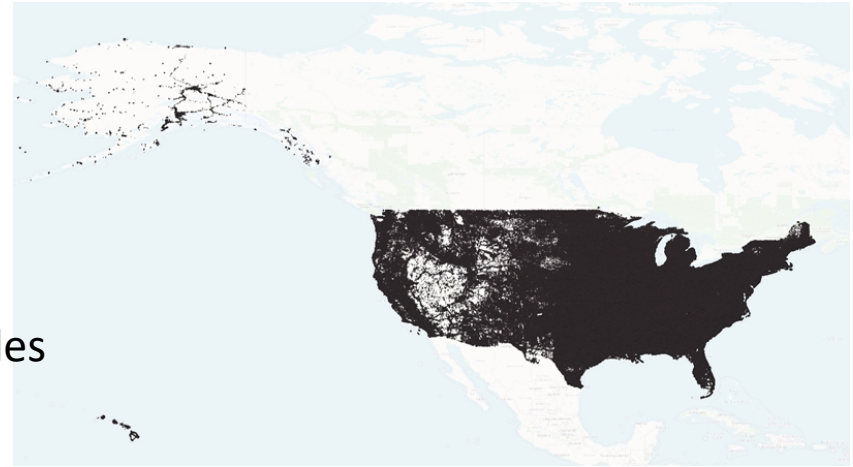


Converted by OSM2GMNS



US rail network from OSM with 0.5M nodes

Handling Large-scale map



US driving network from OSM with 20M nodes

AMS Data Hub Software Prototype: NeXTA

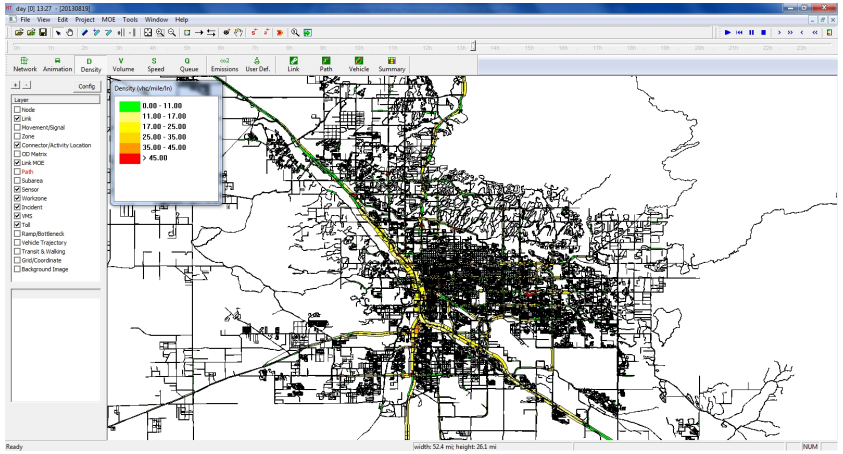
Start with GIS with multiple transportation layers

- 1. node,
- 2. link,
- 3. agent,
- 4. trajectory,
- 5. signal timing.

https://github.com/xzhou99/NeXTA-GMNS

Focus on data processing and visualization

https://www.fhwa.dot.gov/publications/research/operations/13036/004.cfm



The Effective Integration of Analysis, Modeling, and Simulation Tools

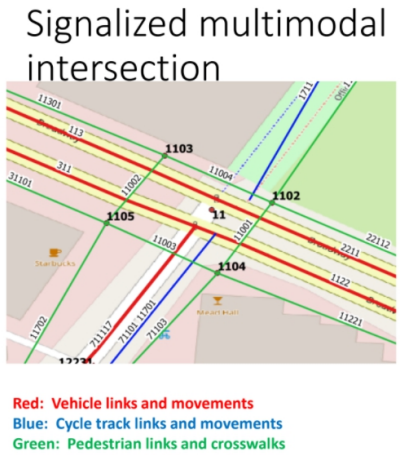
AMS DATA HUB CONCEPT OF OPERATIONS

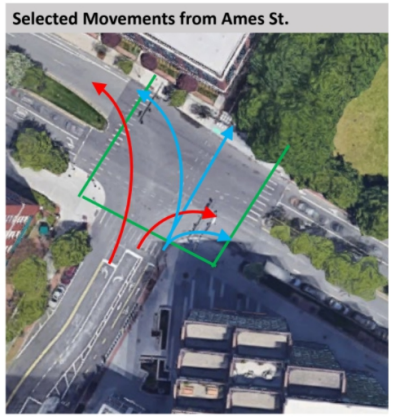
0.3 Signal Timing Generation Tool

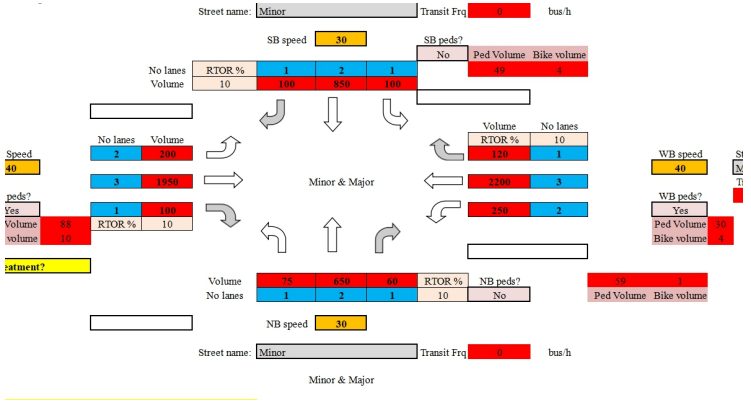
Excel-based and Python based

https://github.com/milan1981/Sigma-X

HCM-based computational engine for signalized intersections







0.4 Demand Generation

Grid2Demand

https://pypi.org/project/grid2demand/

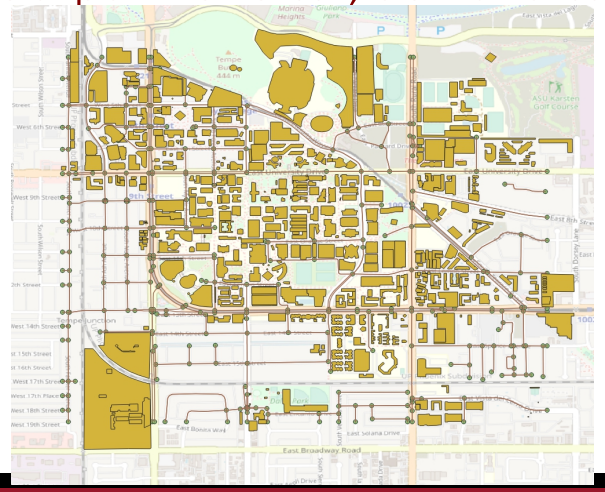
Input: POI geometry production, attraction ir type

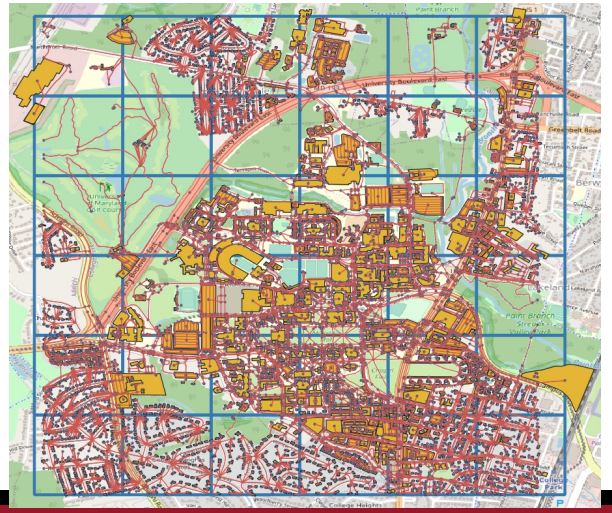
Intermediate: accessibility matrix

Output: zone.csv, demand.csv

$$T_{ij} = P_i \cdot \frac{A_j \cdot F_{ij} \cdot K_{ij}}{\sum_j (A_j \cdot F_{ij} \cdot K_{ij})}$$

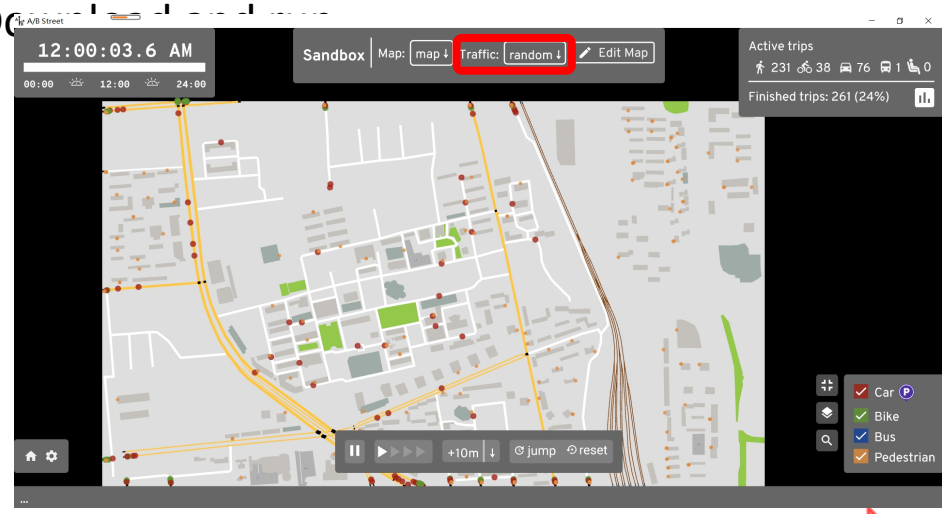
$$F_{ij} = e^{\beta d_{ij} - g}$$





Simulation/Assignment --> Learn Other Educational Resource From ABD30

Do



Select map

Educational Resources

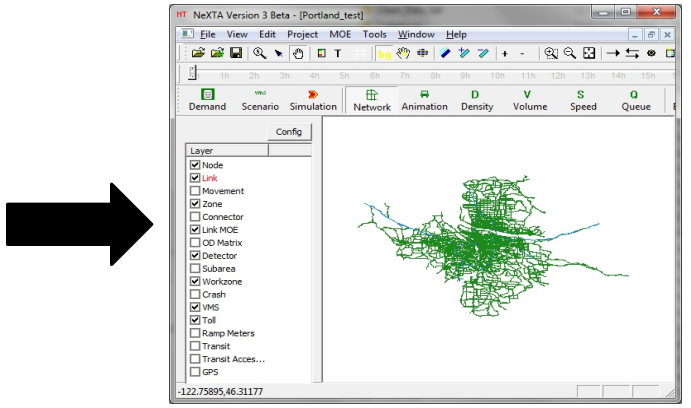
This webpage provides a list of educational resources suggested by our committee members and friends. If you please fill this form, or contact Dr. Steve Boyles.

General Resources

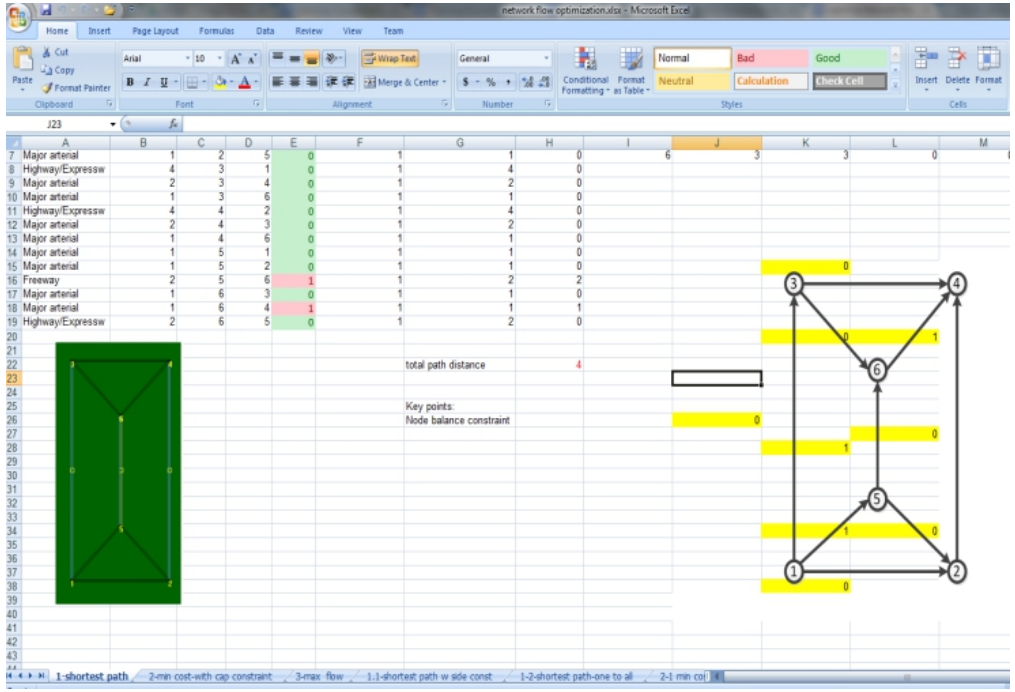
- Transportation Network Analysis, a free textbook written by Stephen Boyles (The University of Texas at Au and Avinash Unnikrishnan (Portland State University). Currently, Volume I (static and dynamic traffic assigr
- Python tutorial for civil engineering students, developed by Avinash Unnikrishnan (Portland State Univers at Austin).
- Transportation Network Analysis graduate course, with online slides and assignments. Covers basics of ne Developed by Stephen Boyles (The University of Texas at Austin).
- Traffic Engineering course materials, developed by Sabyasachee Mishra. (University of Memphis)

General Transit Feed Specification (GTFS)

- Import GTFS to node/link data to GMNS data format (NeXTA-> menu -> import GTFS)

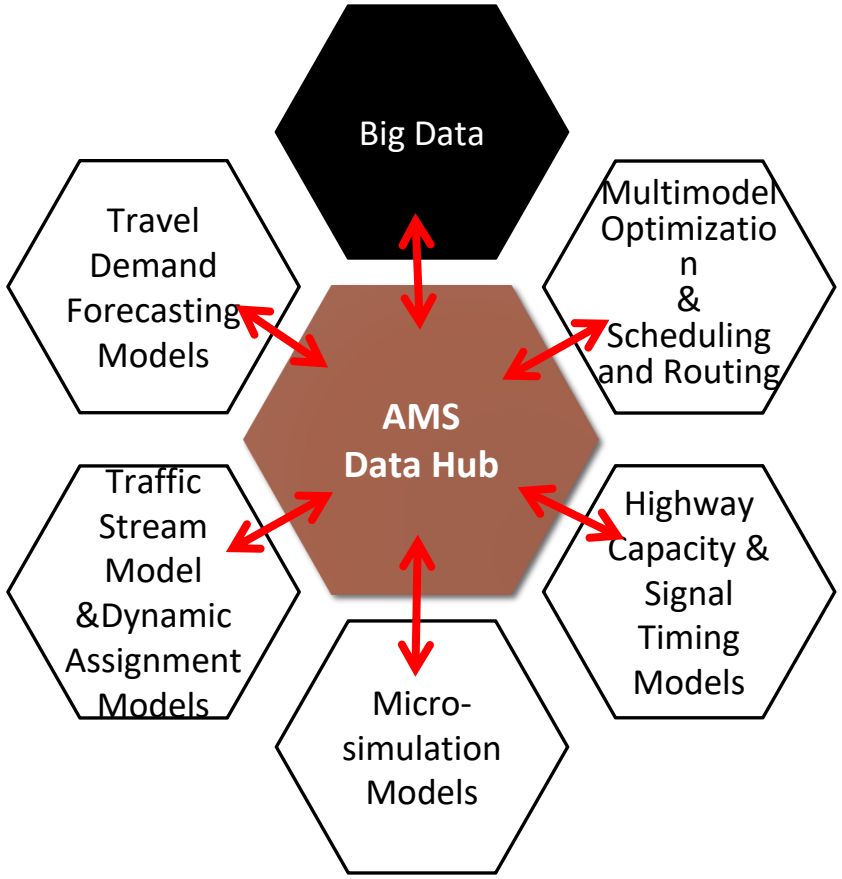


Use of network data:
Shortest Path Finding, Network Flow Model



https://github.com/xzhou99/learning-transportation/blob/master/GAMS_code%20-space-time-network/Assignment_2_LearningNetworkFlowOptimizationinExcelandGLPKSolvers.docx

Connection 1: Big Data

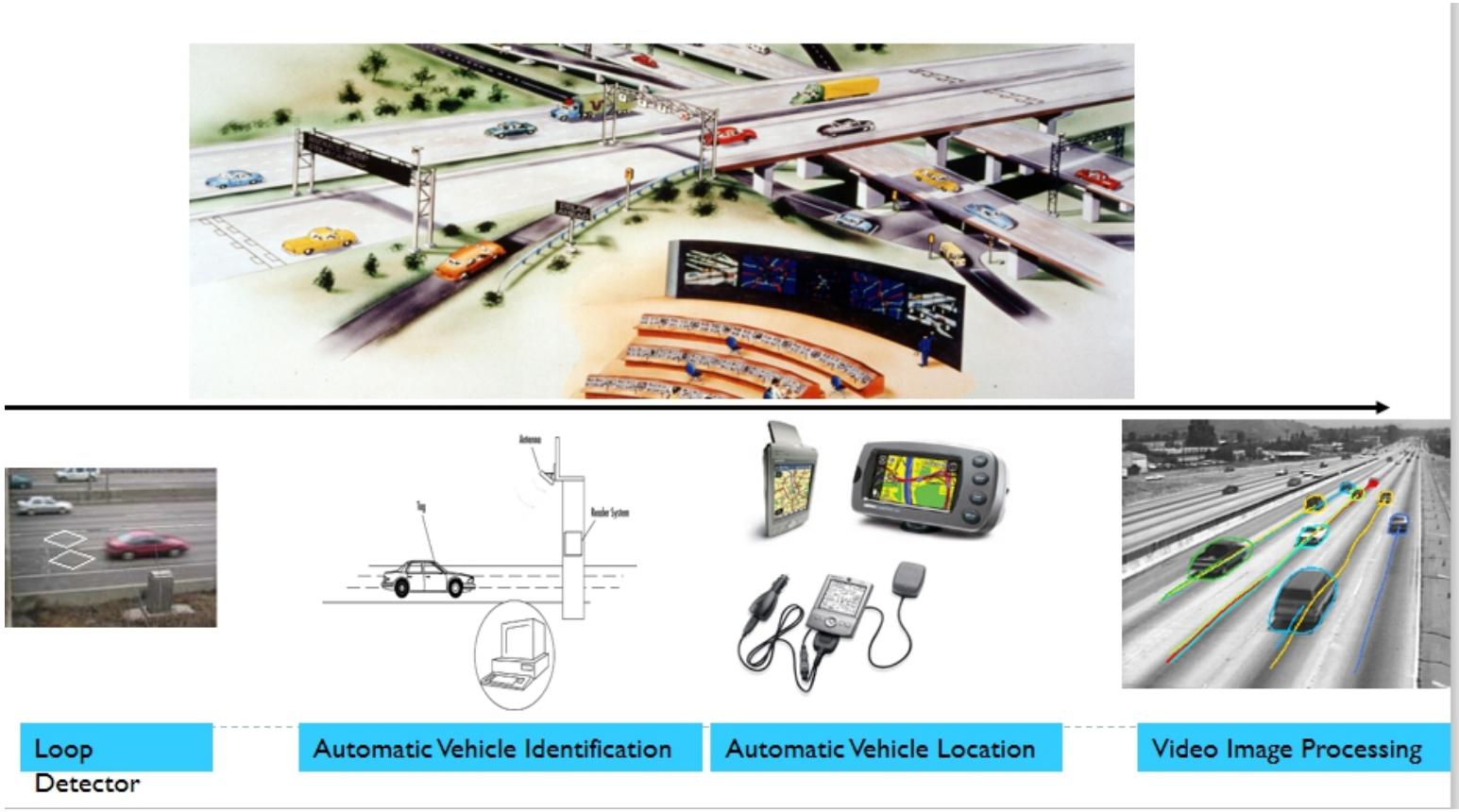


- Sensor data (link_performance.csv)
  15-min speed or count
- GPS trajectory data (trajectory.csv)
- NGSIM trajectory data set
  https://github.com/xzhou99/NeXTA_4_NGSIM_Trajectory_Visualization

BHL sensor data set

I-405 data set from PeMS (to do)

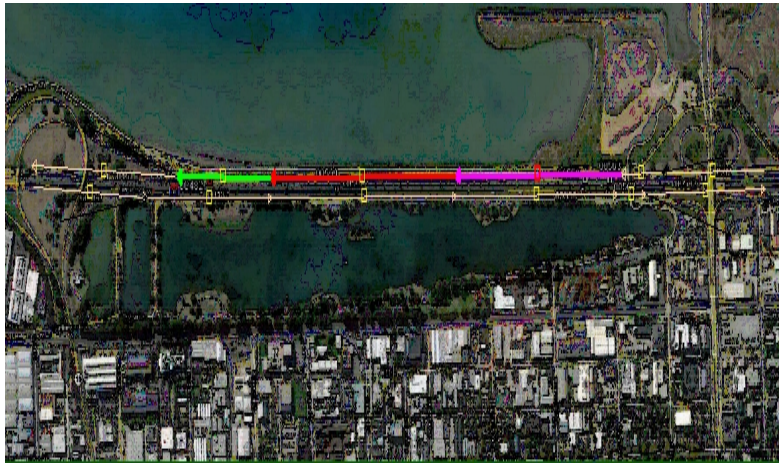
Big data for integrated traffic management

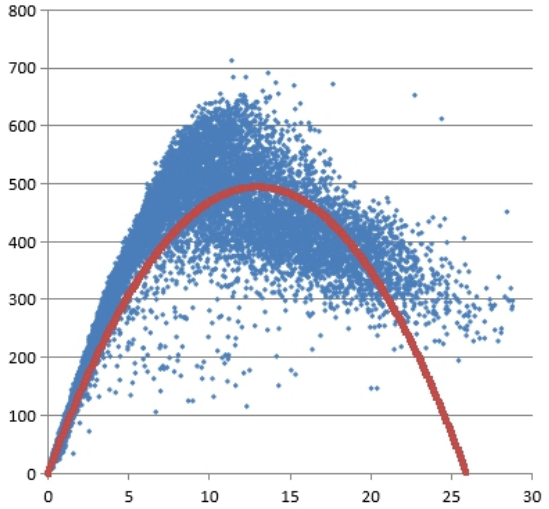


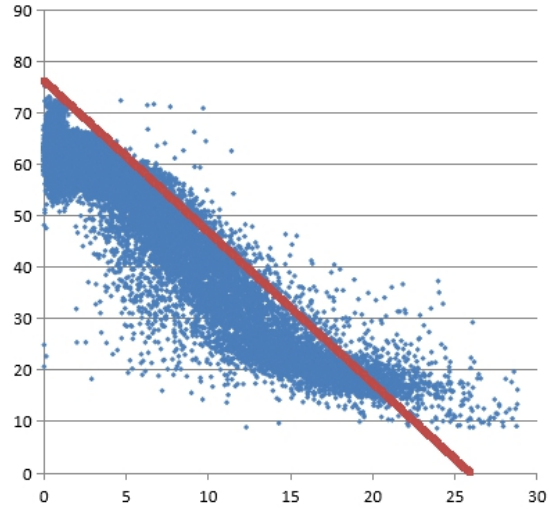
1.1 Understand QKV relationship from aggregated sensor data

Berkeley Highway Laboratory (BHL) sensor data set

https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/4_highway_capacity_traffic_stream_model/BHL-Network

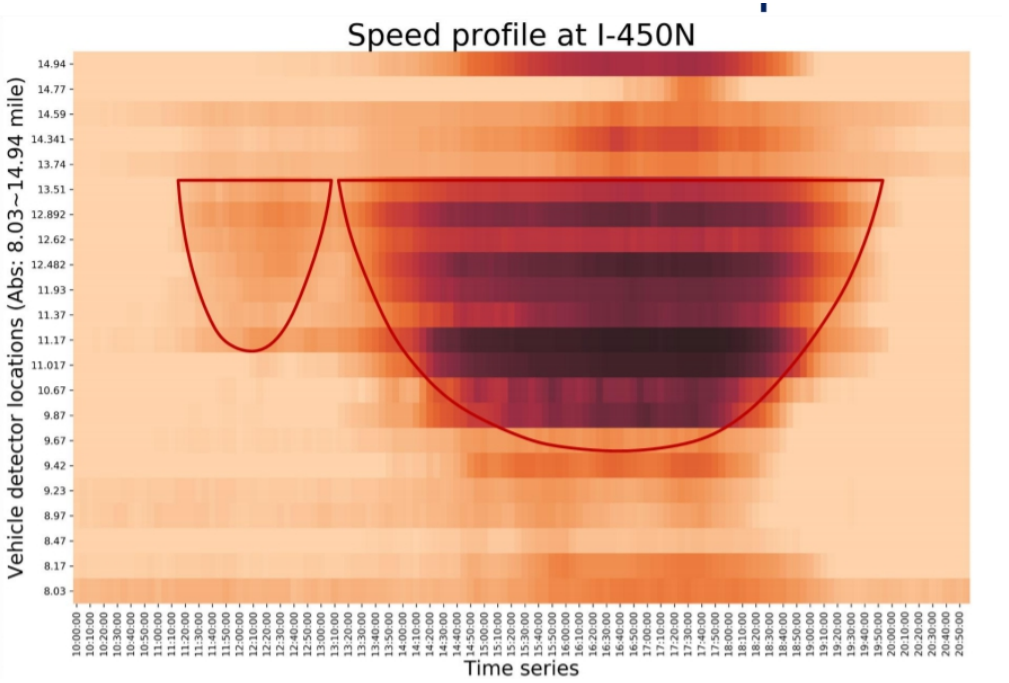


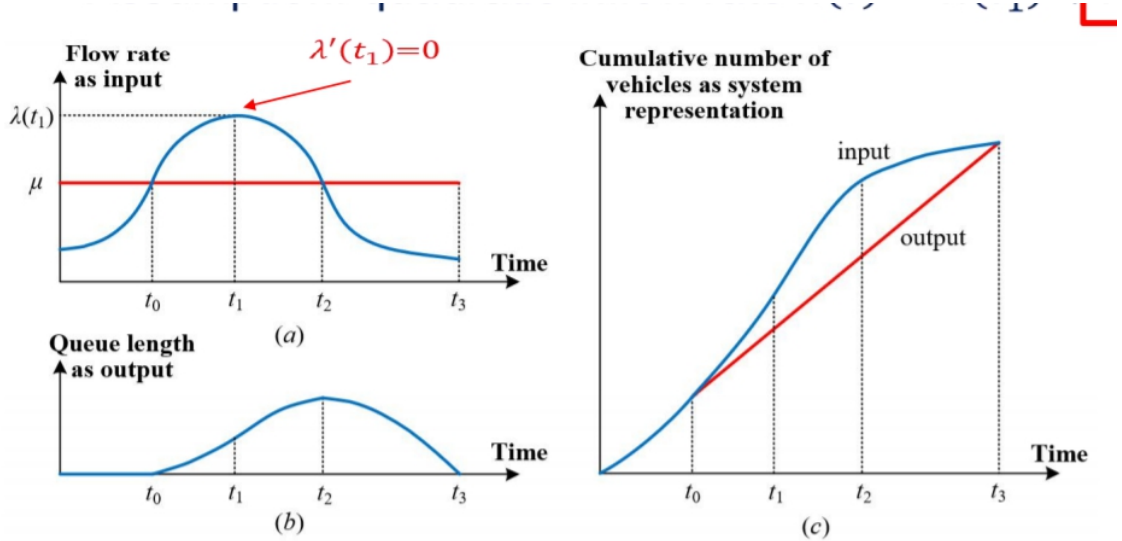




1.2 Understand capacity and bottleneck

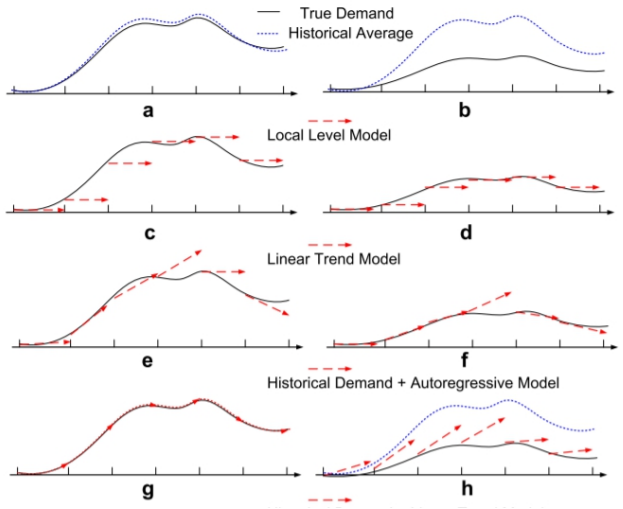
https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/4_highway_capacity_traffic_stream_model

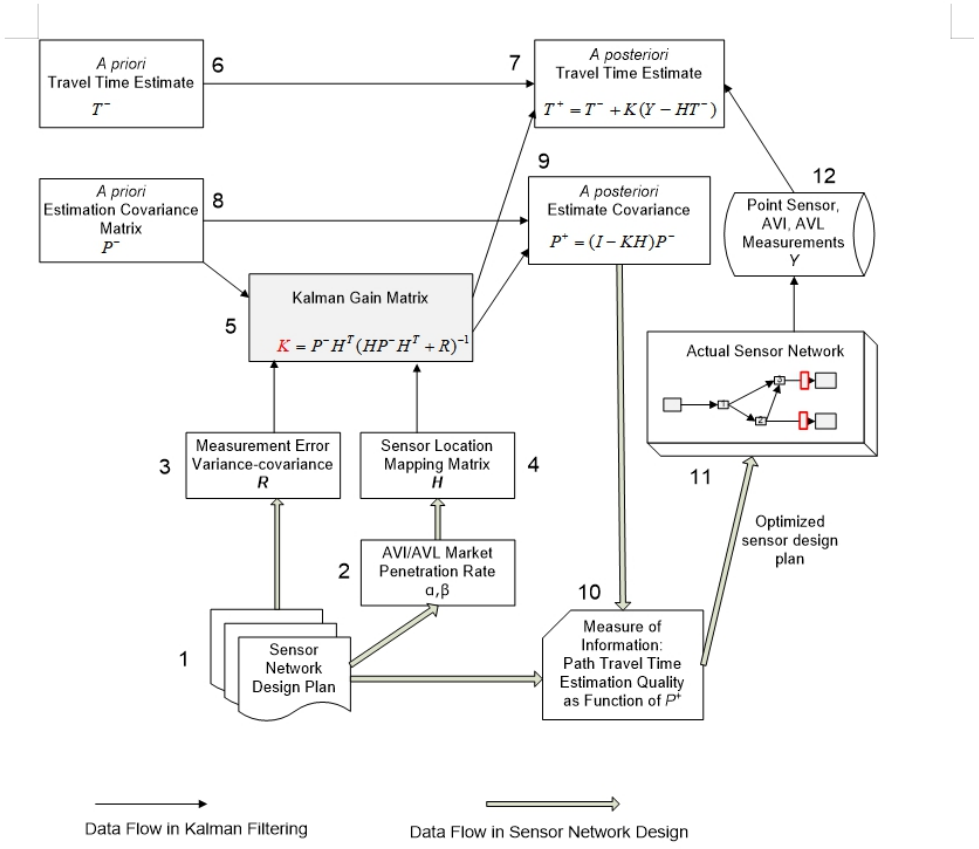




1.3 Understand traffic state estimation and prediction

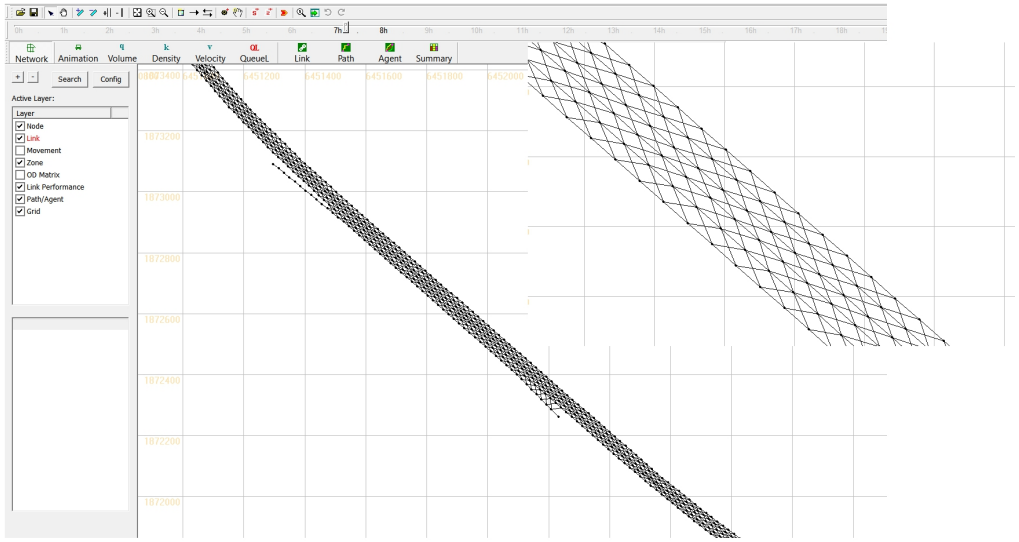
https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/4_highway_capacity_traffic_stream_model/Learning_KalmanFiltering_for_travel_state_estimation_prediction

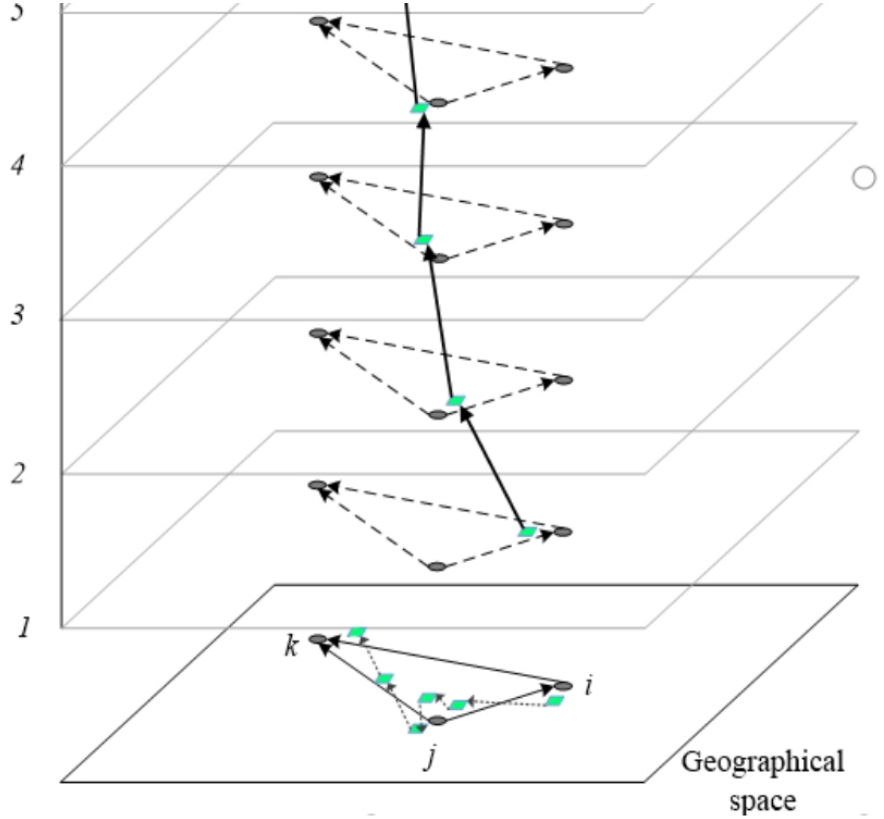




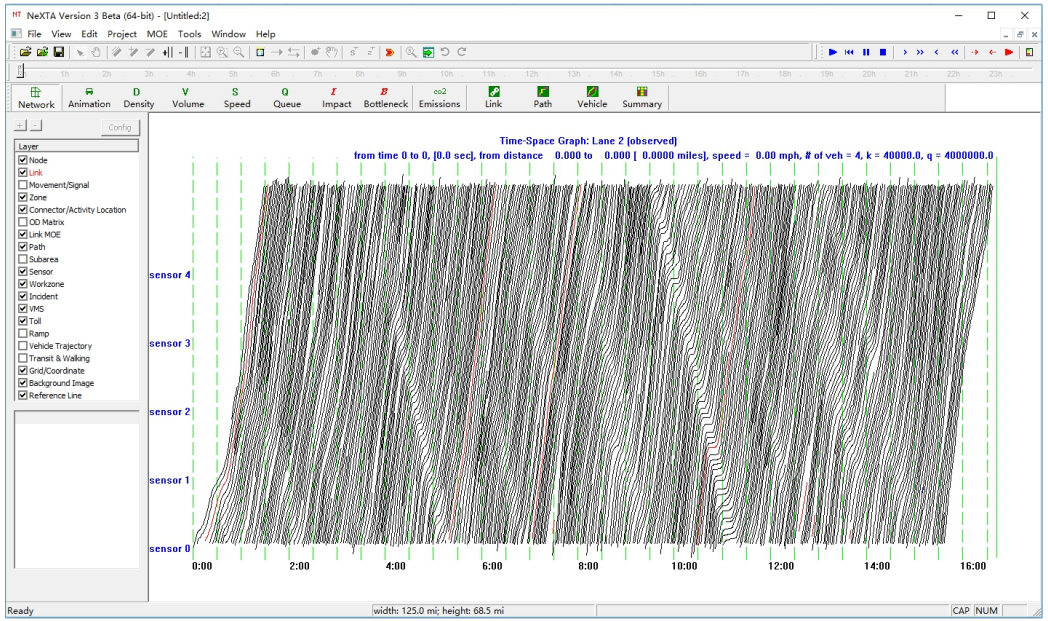
1.4 Map matching based on space-time paths and a time geographic method

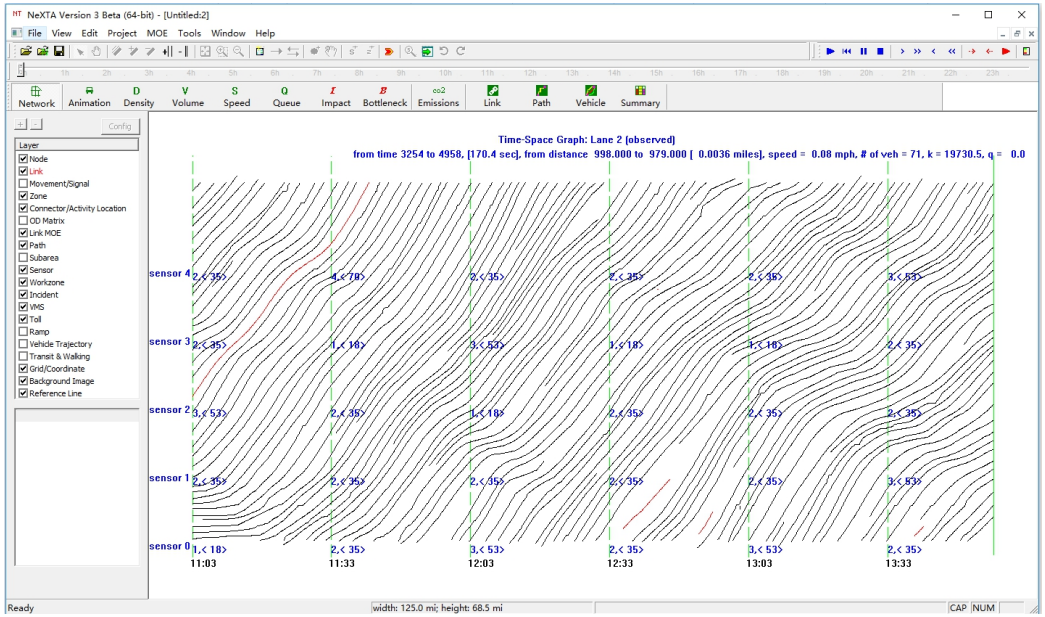
https://github.com/xzhou99/space-time-network-based-map-matching





1.5 Analysis of Trajectories from NGSIM

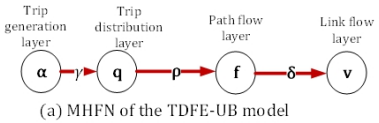




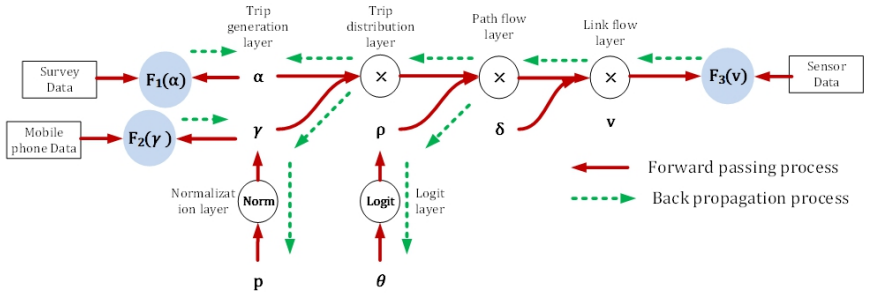
1.6 Integrate deep learning methods with transportation modeling

https://github.com/xzhou99/TCGLite

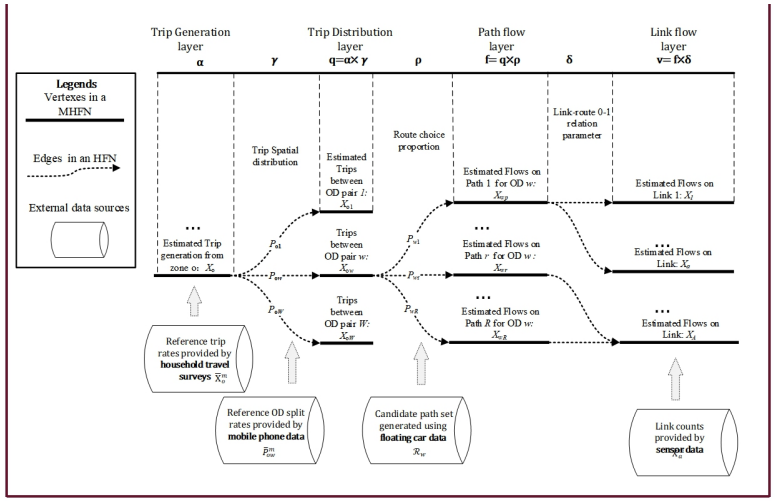
Computational graph language to describe the back propagation



(a) MHFN of the TDFE-UB model



(b) The corresponding computational graph for the MHFN of the TDFE-UB model

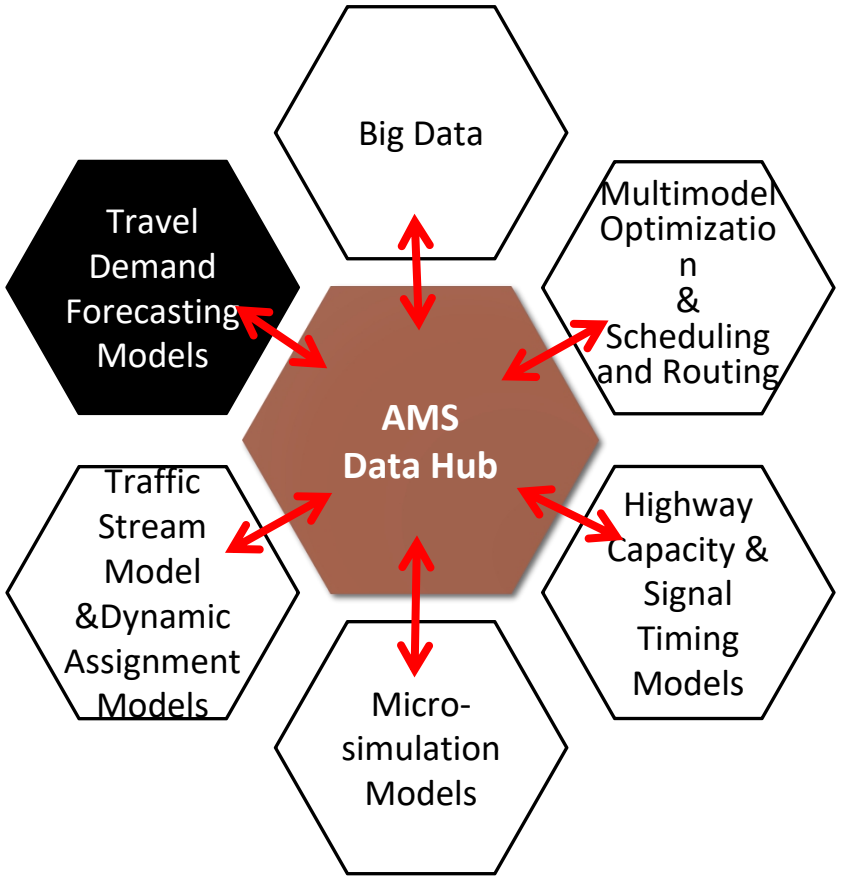


1. **The following flow conservation constraints are expressed by the "Neural Network" whose activation functions are ReLu function** $f(x) = \max(0, x)$:

$$X_o P_{ow} = X_w \quad \forall w \in \mathcal{W}\ o \in \mathcal{Z}$$
$$X_w P_{wr} = X_r \ \forall r \in \mathcal{R}\ w \in \mathcal{W}$$
$$\sum_{r \in \mathcal{R}} \delta_{ra} X_r = X_a \quad \forall a \in \mathcal{A}$$

2. **Different types of data sources are mapped onto different layers of the architecture**

Connection 2: Travel demand model

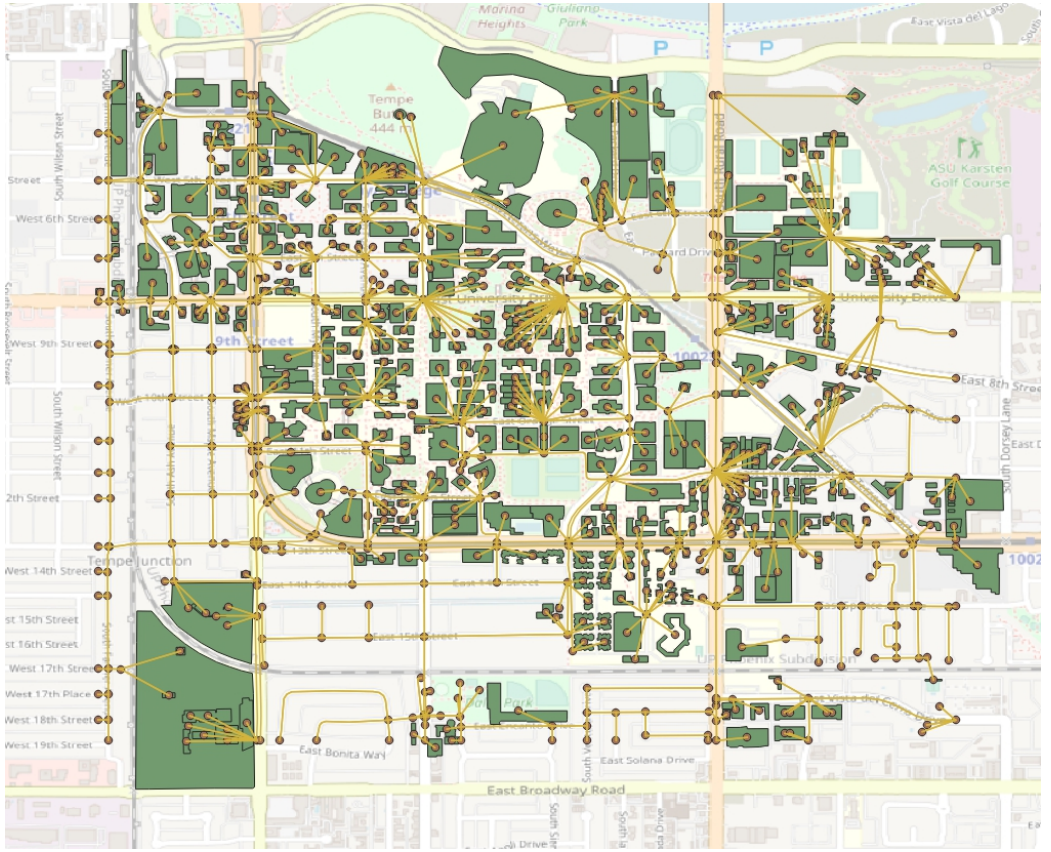


- Trip generation, distribution and model choice
- Static Traffic Assignment
- OD destimation using multiple data sources
- Layout and land use

2.1 Trip generation, distribution and model choice

- Grid2Demand
  - A quick trip generation and distribution tool
  - 4-step trip generation
  - Input: POI geometry production, attraction information, POI type
  - Output: zone.csv, demand.csv
  - https://github.com/asu-trans-ai-lab/grid2demand

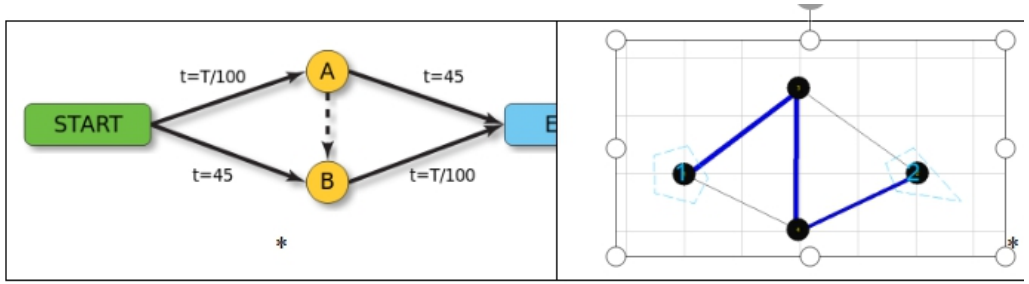
- Mode choice model
  - https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/6_trip_generation/mode_choice



POI mapping from OSM

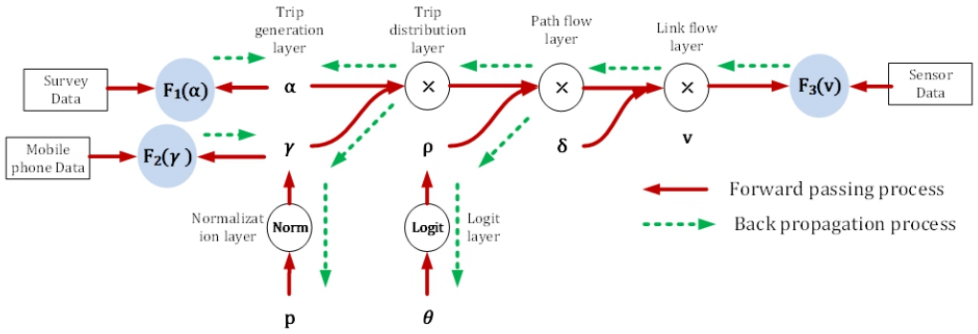
2.2 Static Traffic Assignment

- Static traffic assignment GAMS
  - https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network/2%20traffic_assignment
- Static traffic assignmetn using STALite
  - https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/2_static_user_equilibirum

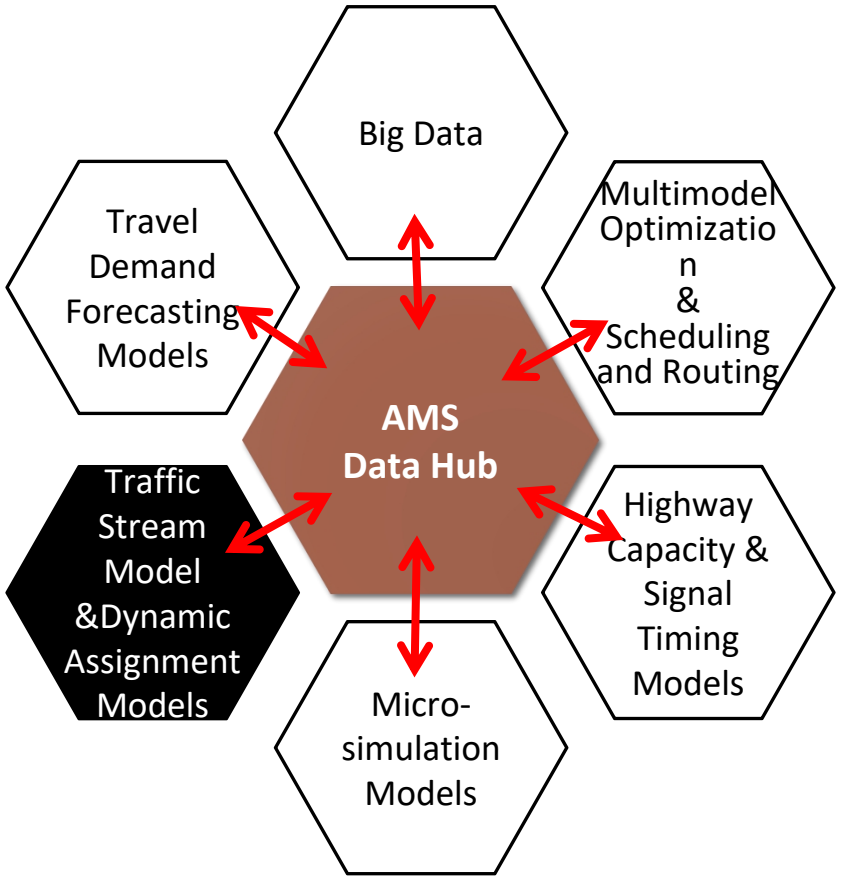


2.3 OD Demand Estimation

- Static OD demand estimation using STAlite
  - https ://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/7_ODdemand_estimation
- ODME using GAMS
  - https ://github.com/xzhou99/ODME
- Computational graph based demand estimation model using multiple data sources
  - https://github.com/xzhou99/TCGLite

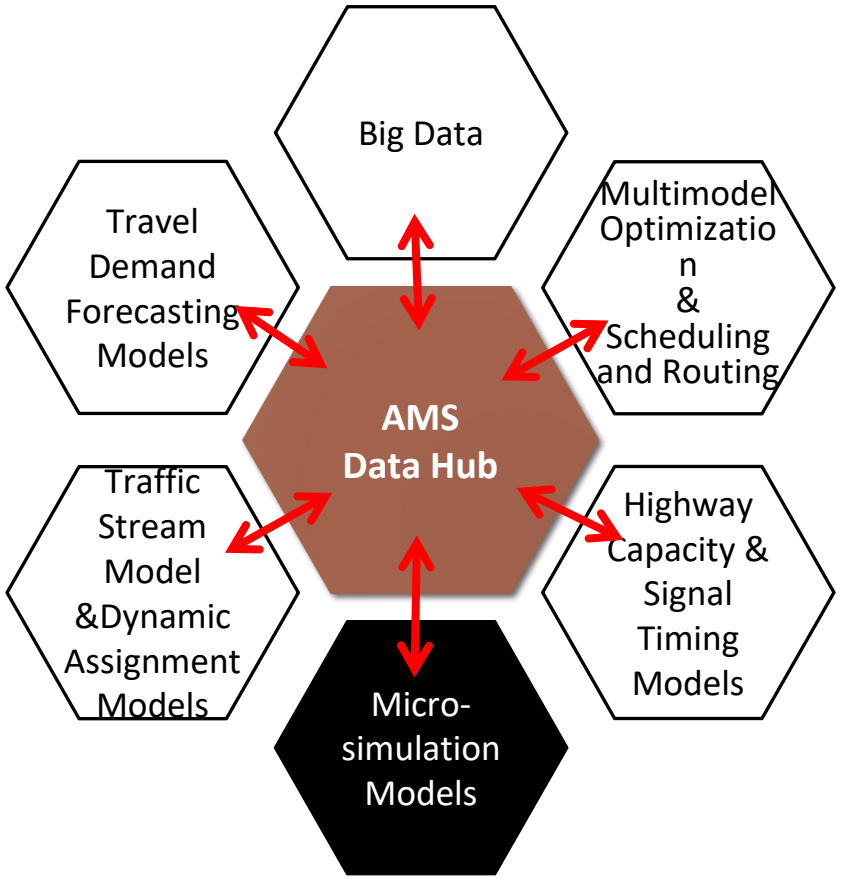


Connection 3: Traffic Stream Model and Dynamic Assignment



- Dynamic user equilibrium assignment method
  - Model queue spillback
  - Consider traffic flow dynamics
- GAMS for dynamic traffic assignment
  - https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network/2%20traffic_assignment
- DTALite/STALite
  - https:// github.com/xzhou99/DTALite-python
  - https://github.com/xzhou99/stalite-dtalite_software_release
  - Learning document
  - https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/3_dynamic_user_equilibrum_and_traffic_bottleneck

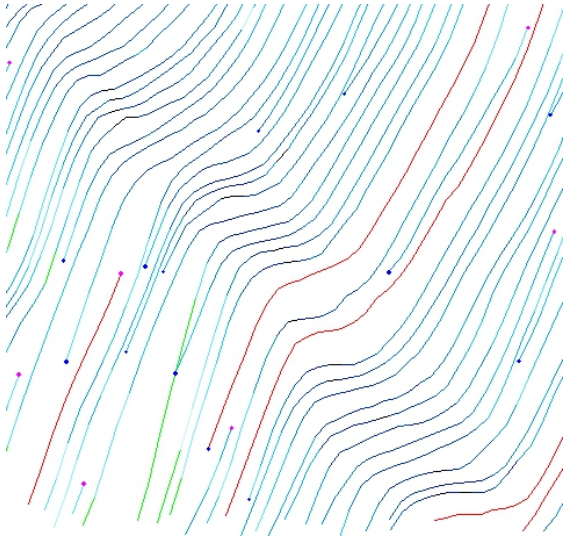
Connection 4: Microsimulation Models

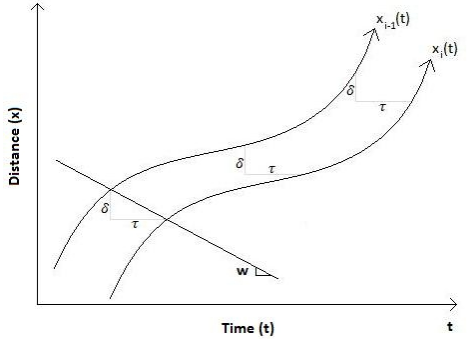


- https://github.com/xzhou99/NeXTA_4_NGSIM_Trajectory_Visualization
- Multi-resolution network convertoer: Ocean https://github.com/jiawei92/Ocean
- CAVLite https://github.com/jiawei92/CAVLite

4.1 Understand driving behavior

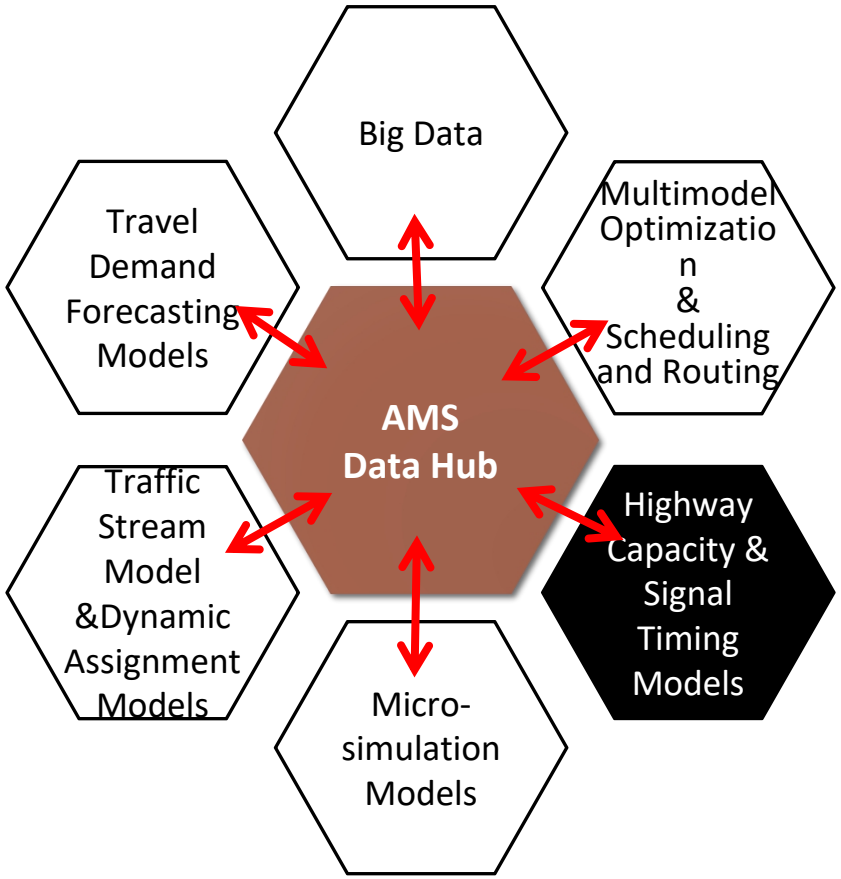
- Underststand microscopic car following model
  - https://github.com/xzhou99/DTW_Vehicle_Trajectory_Data
  - https://github.com/xzhou99/Basic_DTW_Matlab





Reaction distance/spacing δ
Reaction time lag τ
$W = \delta / \tau$

Connection 5: Capacity Analysis Tools



- HCM-based Quick Estimation Method (QEM)
  - Given turning volume,
  - Generate signal phasing and timing data
  - https://github.com/xzhou99/Sigma-X
  - https://github.com/xzhou99/SignalAPI

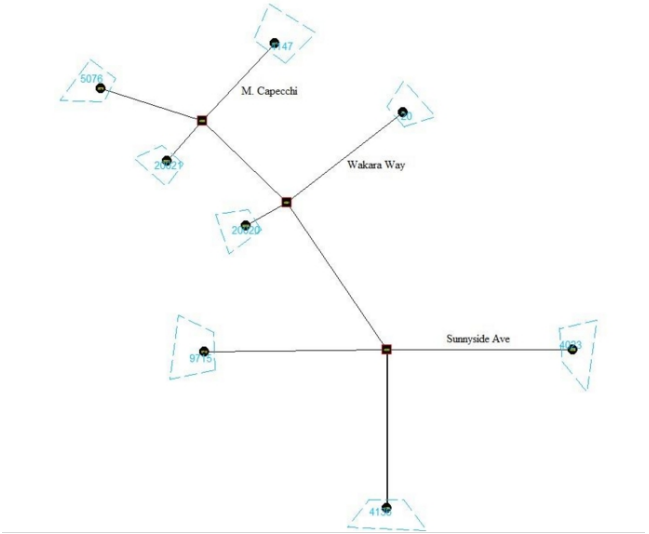
- Learning Document for Signal Timing Optimization
  https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/5_signal_timing_optimization

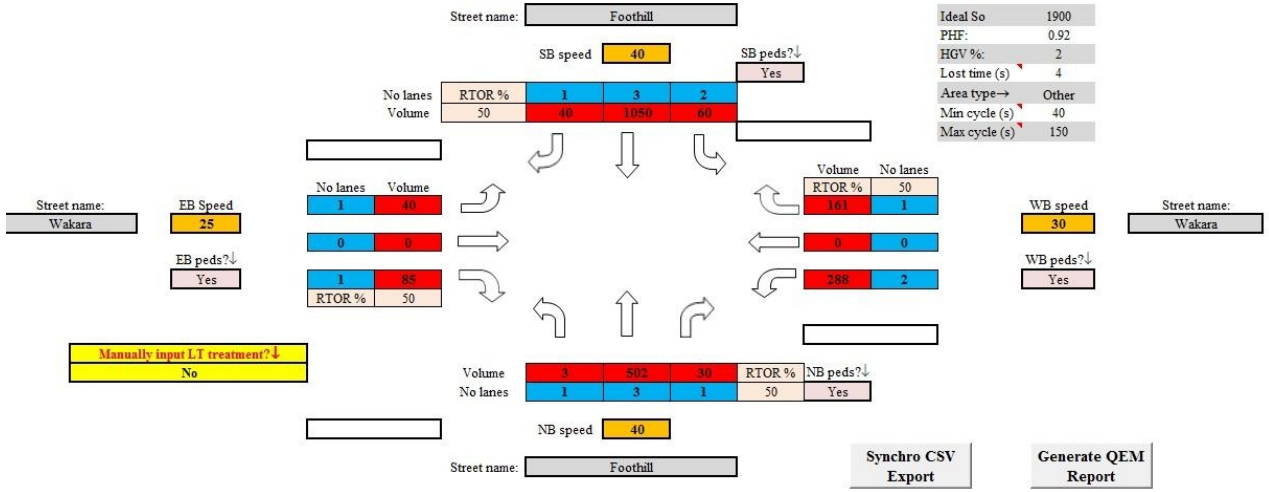
5.1 Traffic signal timing optimization

Excel-based and Python based

- https://github.com/xzhou99/Sigma-X HCM-based computational engine for intersections
- https://github.com/xzhou99/SignalAPI

The Intersection with ID

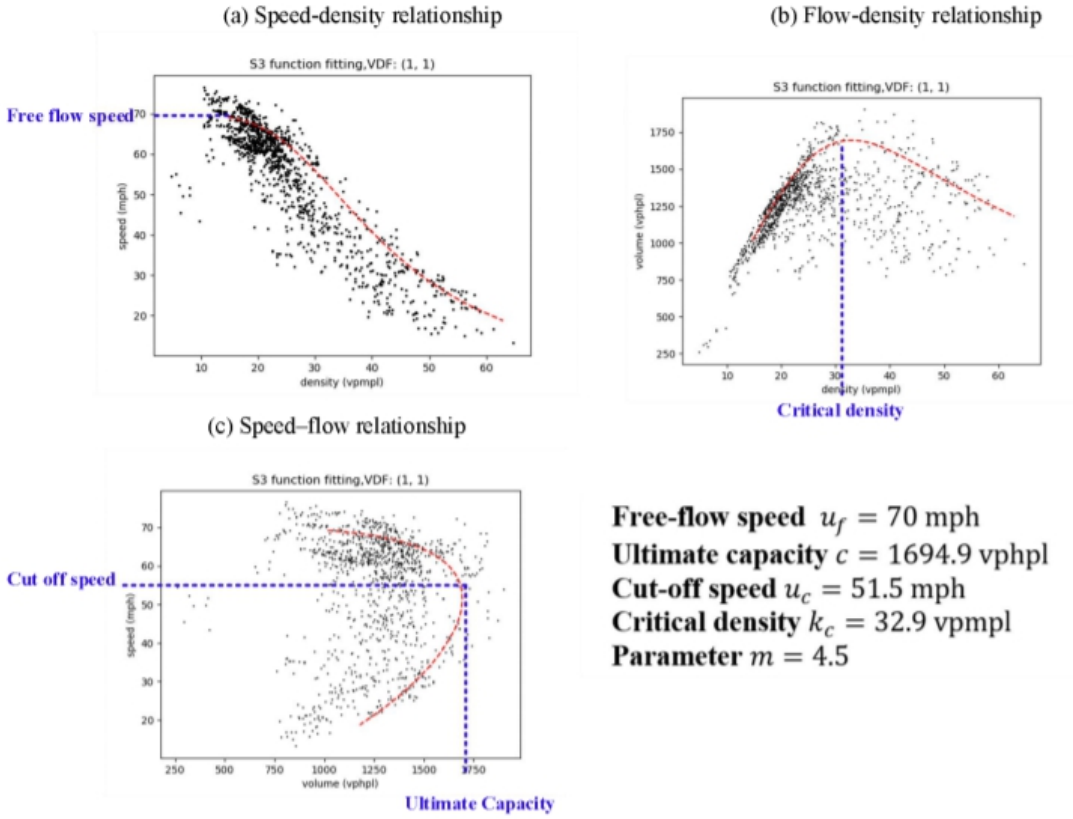




Sigma-X

5.2. Freeway capacity analysis

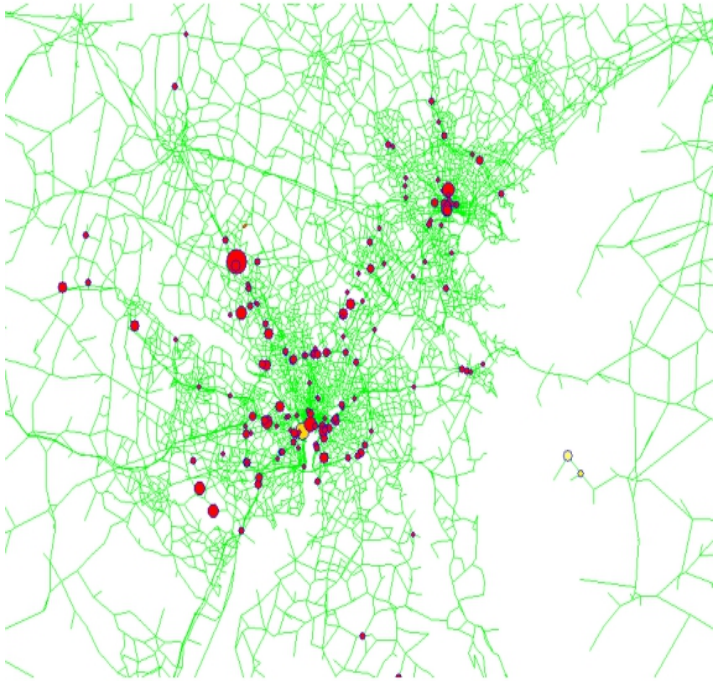
https://github.com/xzhou99/VDF_calibration



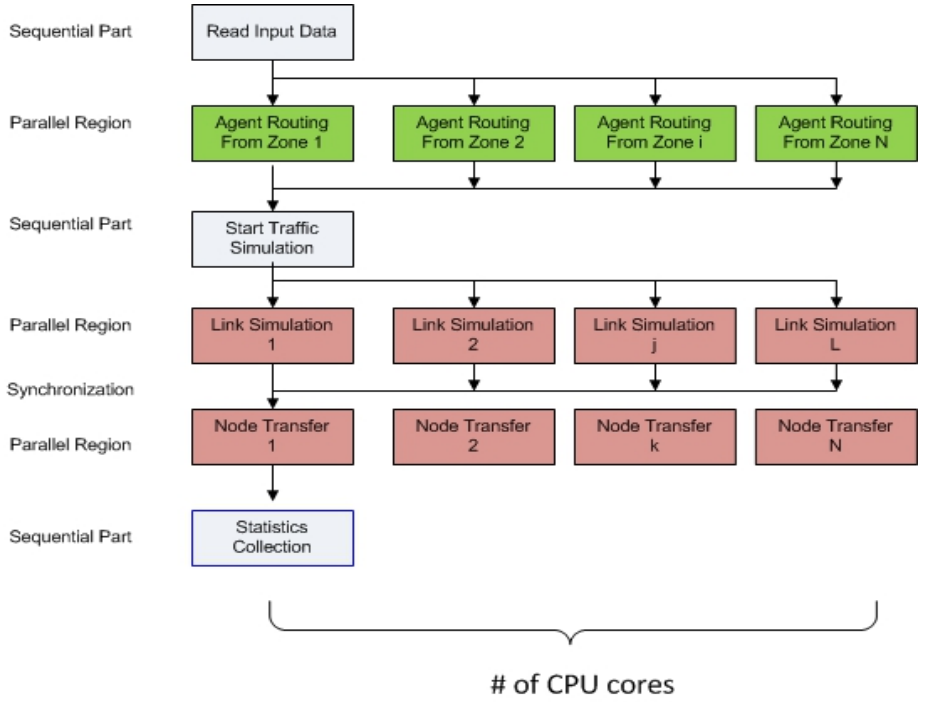
Free-flow speed $u_f = 70$ mph
Ultimate capacity $c = 1694.9$ vphpl
Cut-off speed $u_c = 51.5$ mph
Critical density $k_c = 32.9$ vpmpl
Parameter $m = 4.5$

5. Large scale applications

Computational Challenges

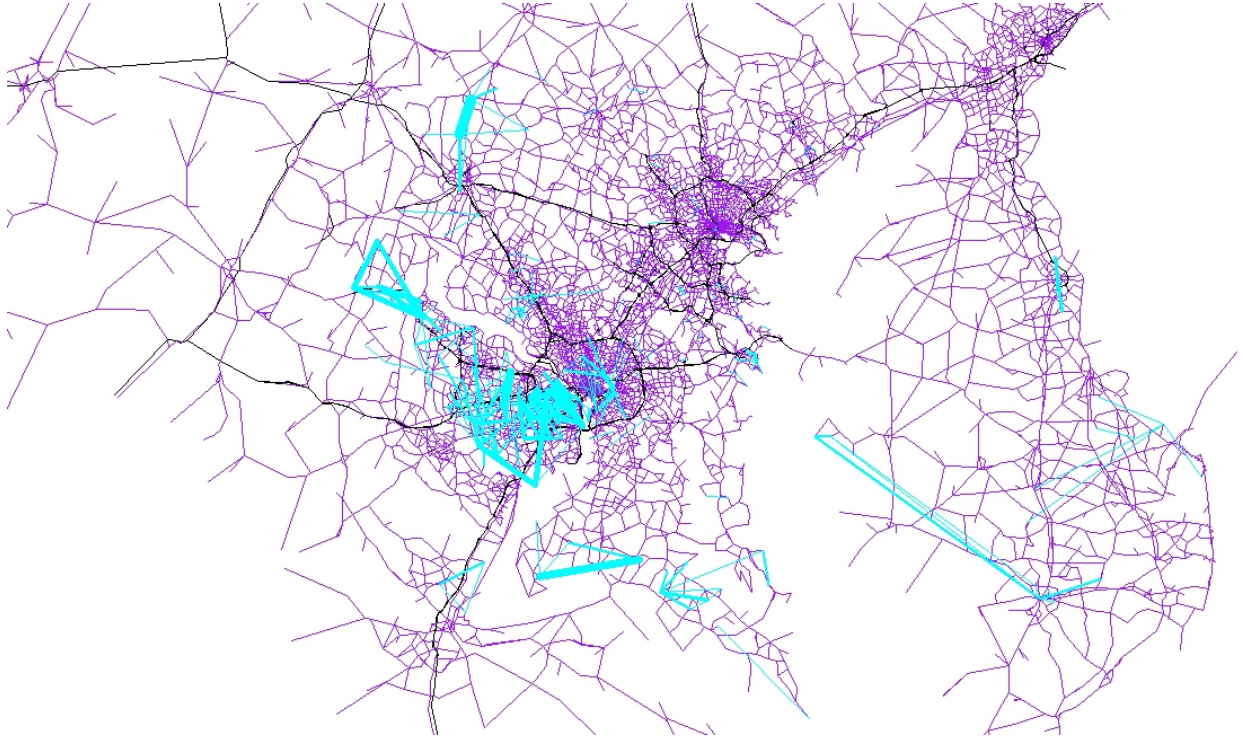


Shared memory-based parallel computing for agent-based path finding and mesoscopic traffic simulation (based on OpenMP)



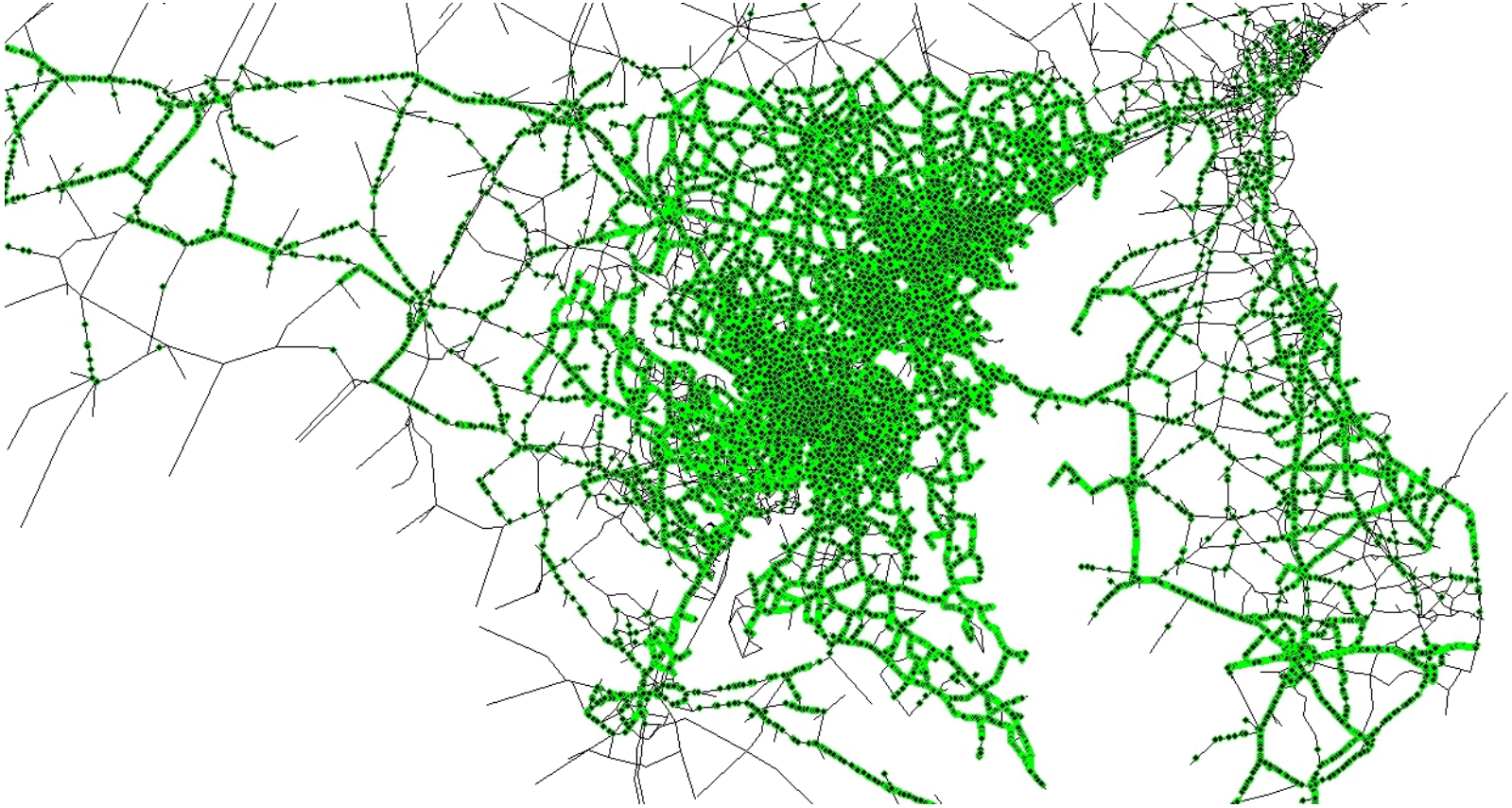
Maryland State-wide model:
20 K nodes, 47K links, 3,000 zones, 18 M agents
CPU time: 30 min per UE iteration on a 20-core workstation with 194 GB RAM

Origin-Destination Demand Spatial Distribution Pattern

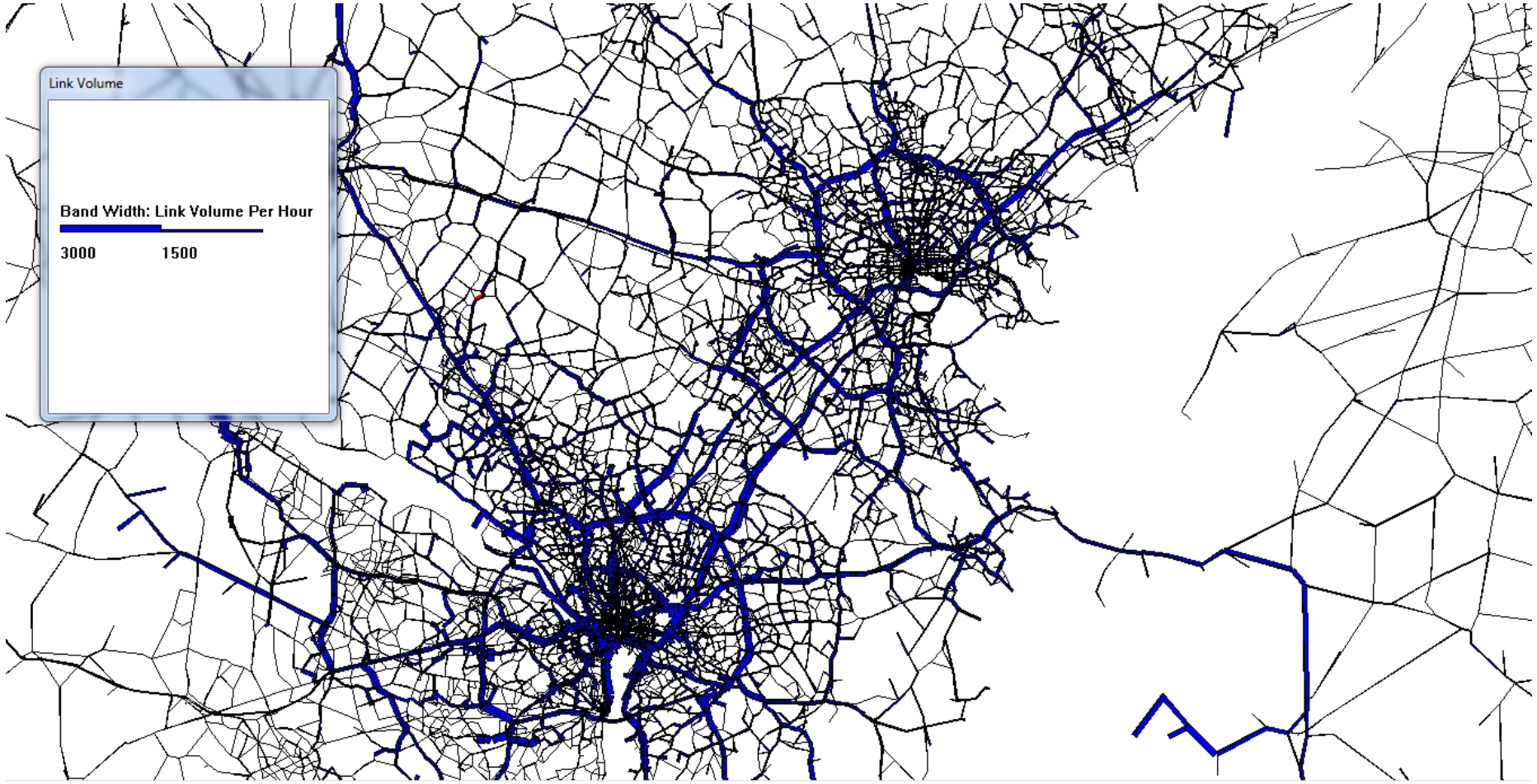


Collaboration with University of Maryland and Maryland State Highway Administration
Supported by TRB SHRP II Program

Vehicle Animation at Network Level

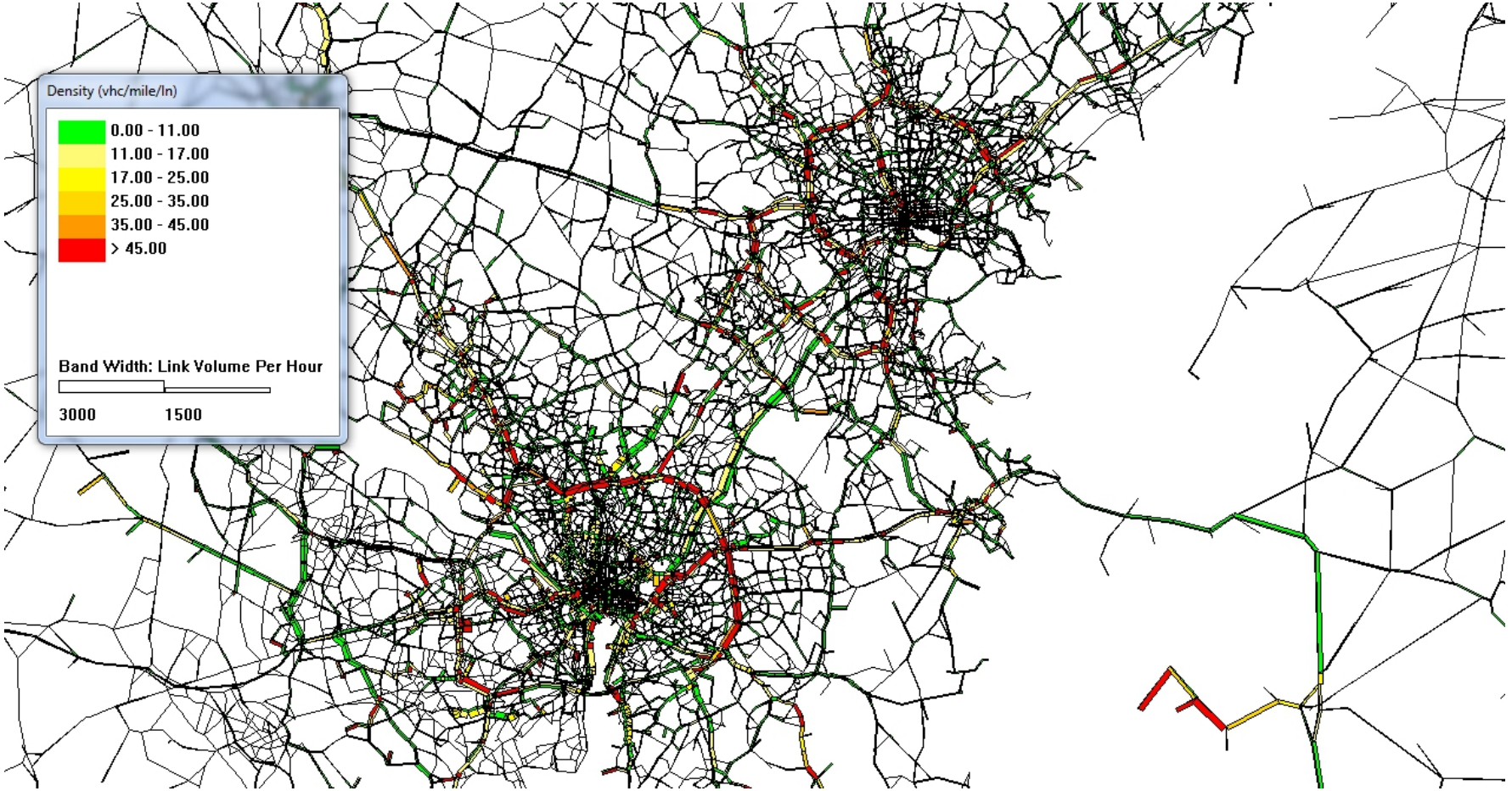


Volume at Network Level

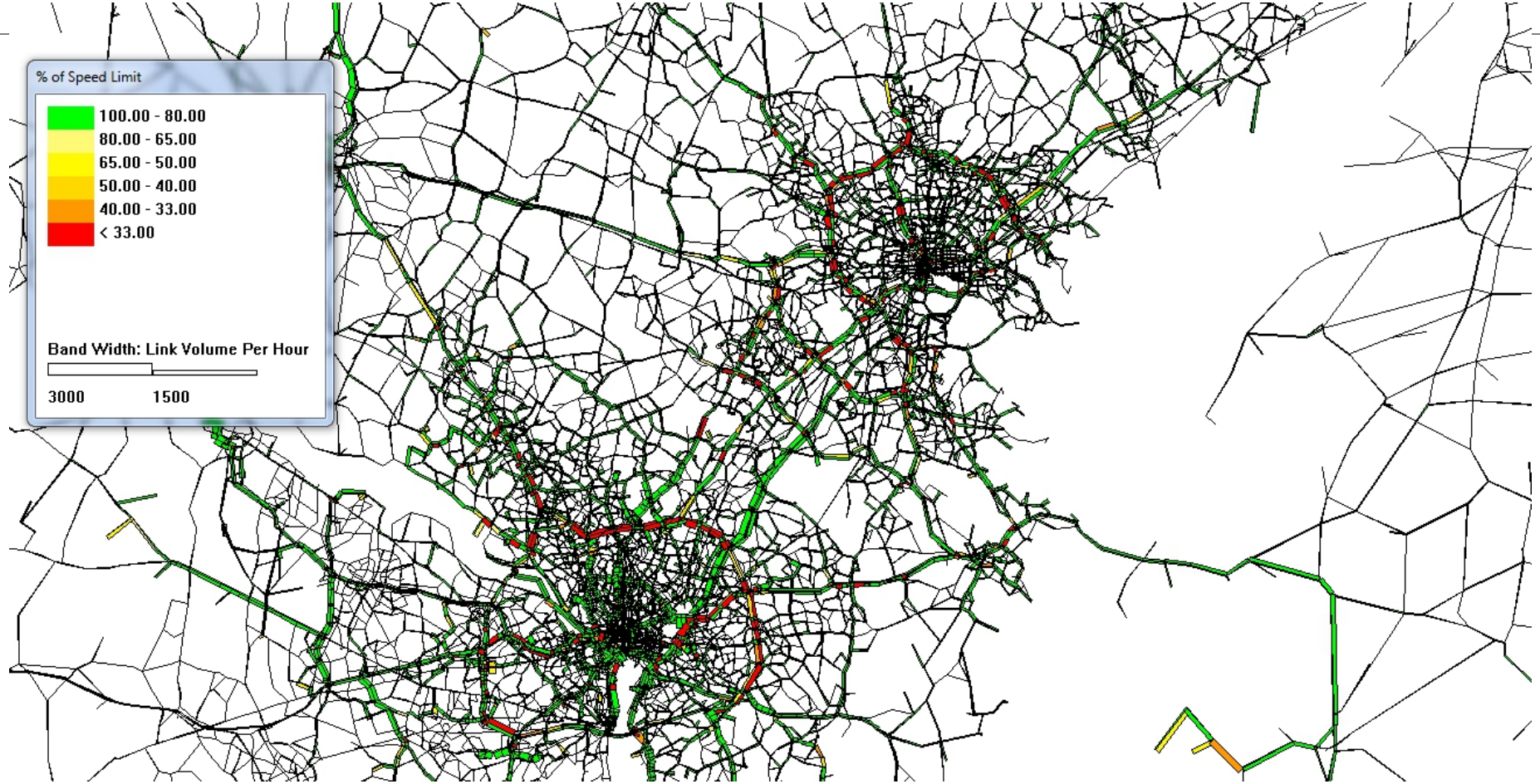


Band width of a link is proportional to link volume

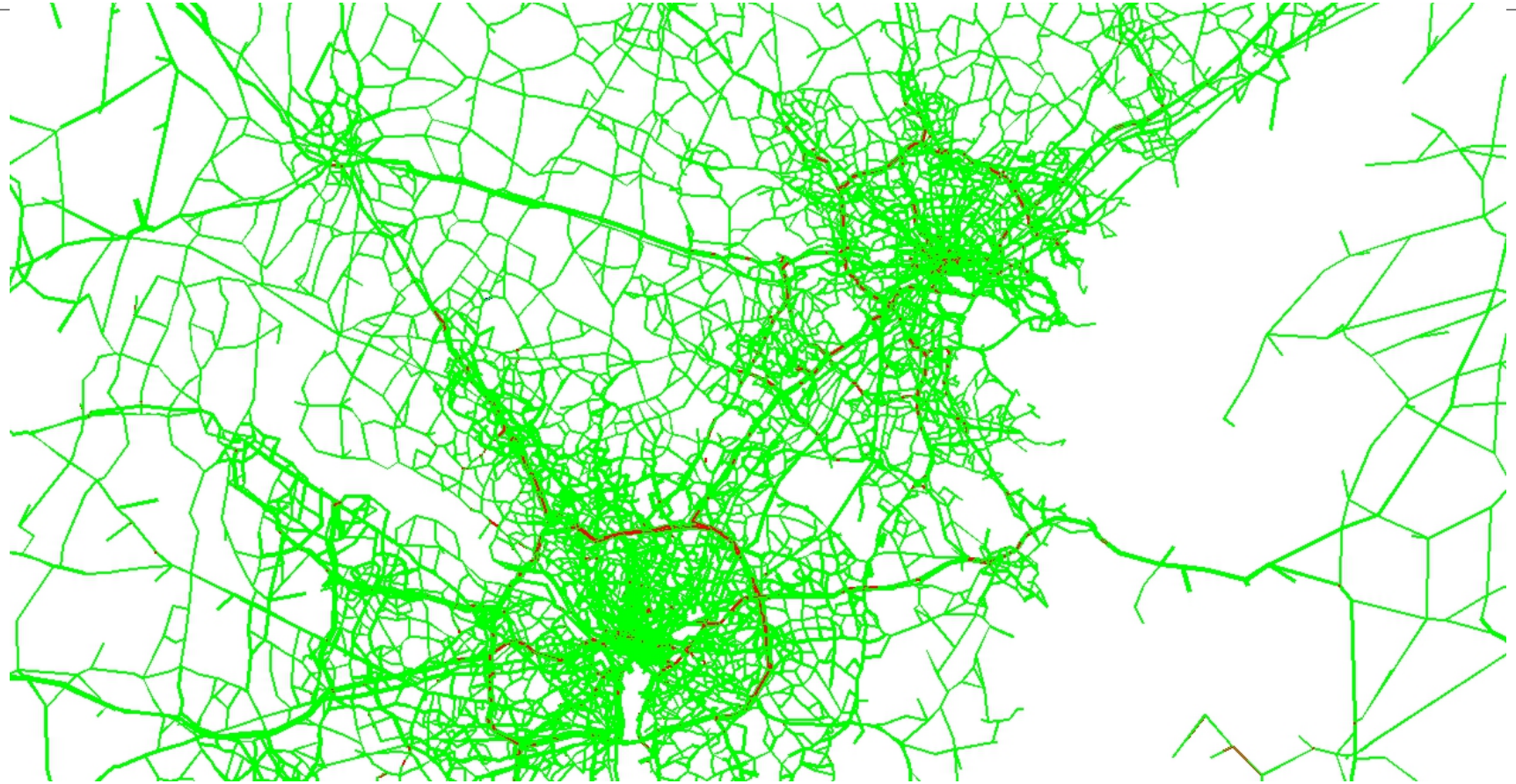
Density at Network Level



Speed at Network Level

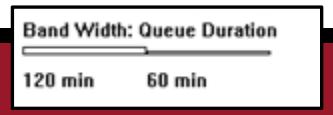


Queue at Network Level

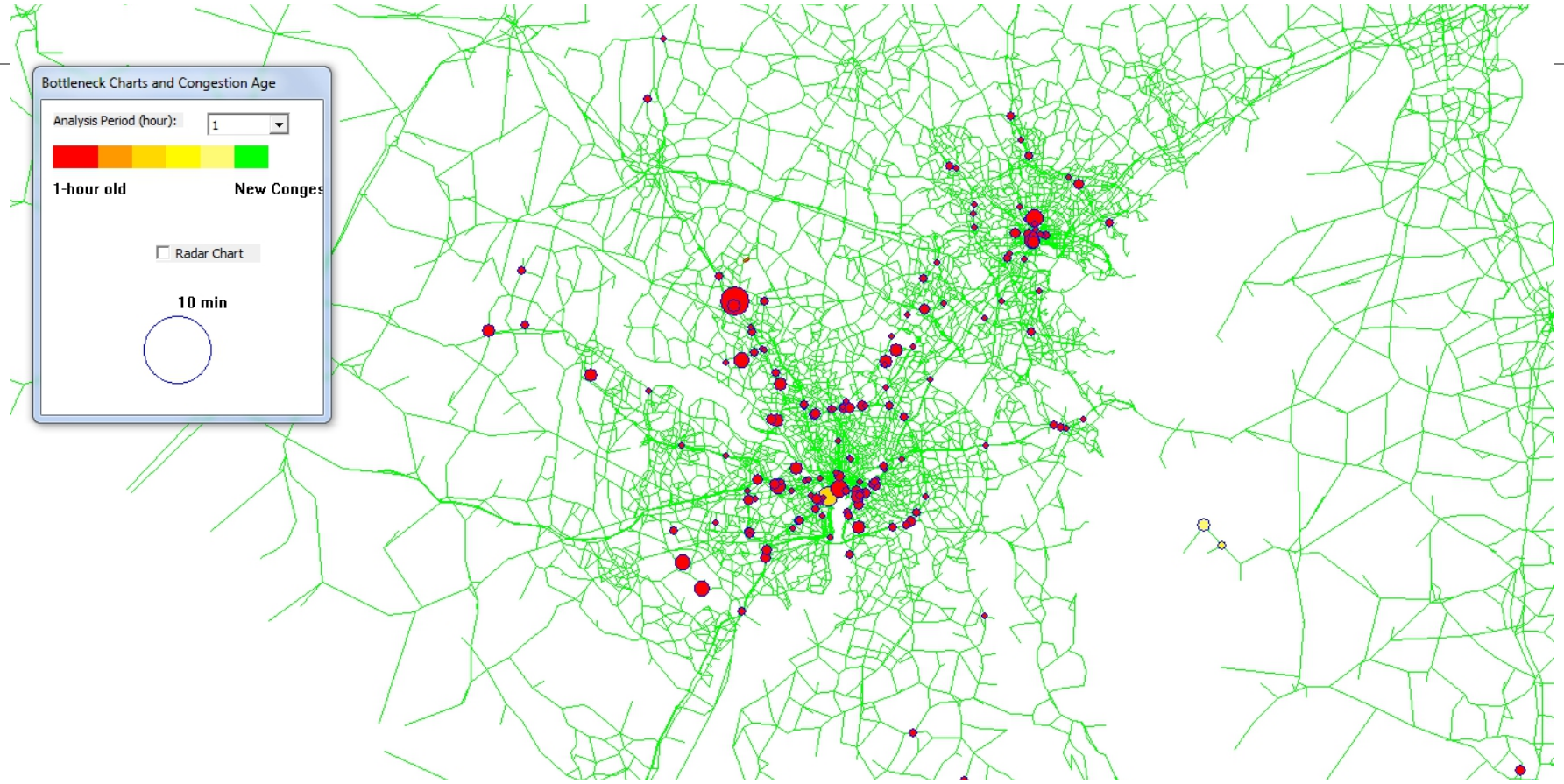


Queue Duration at Network Level

Link width represents duration of congestion (e.g. 60 min vs. 120 min)



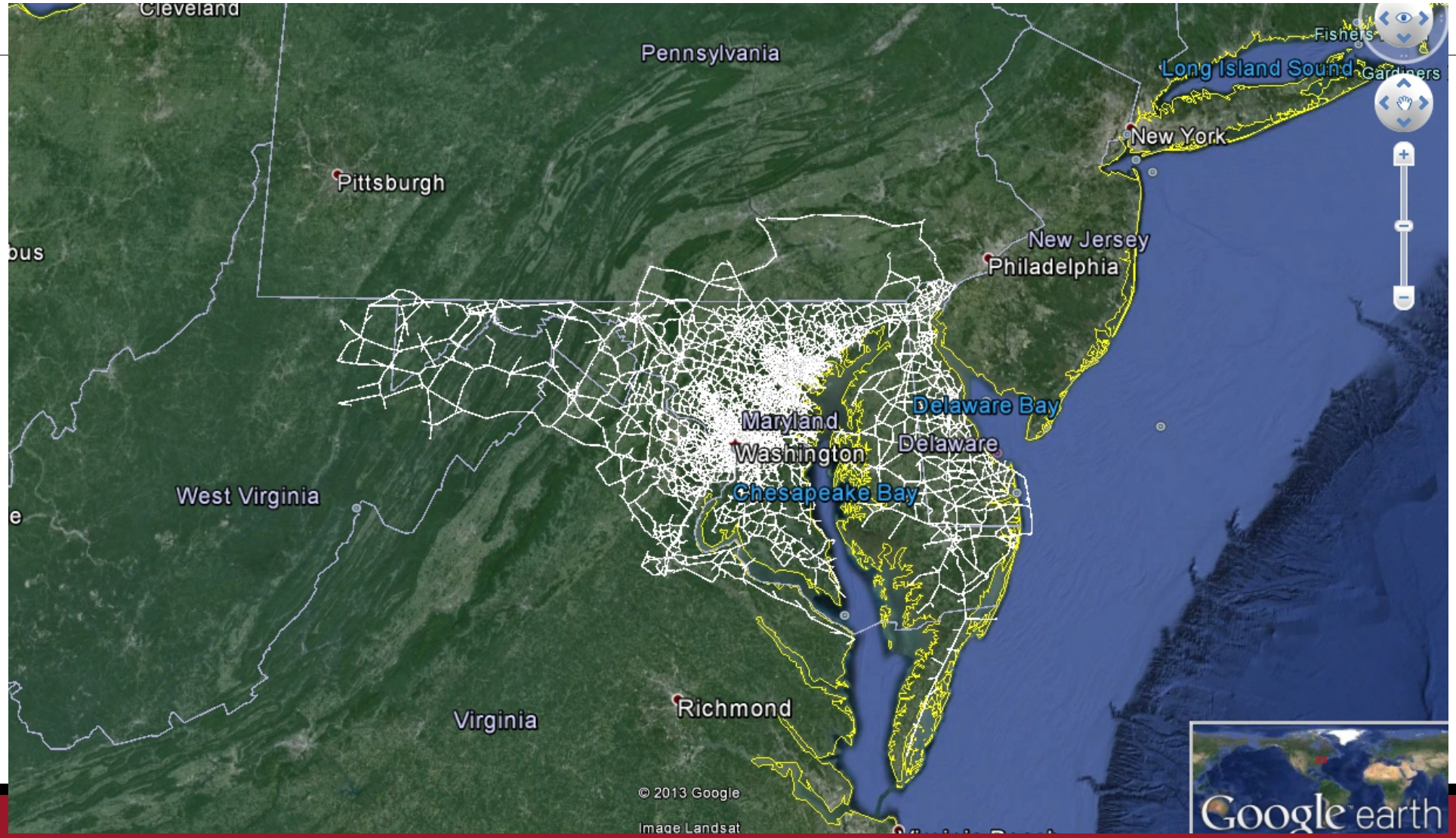
Time-dependent Bottleneck Locations



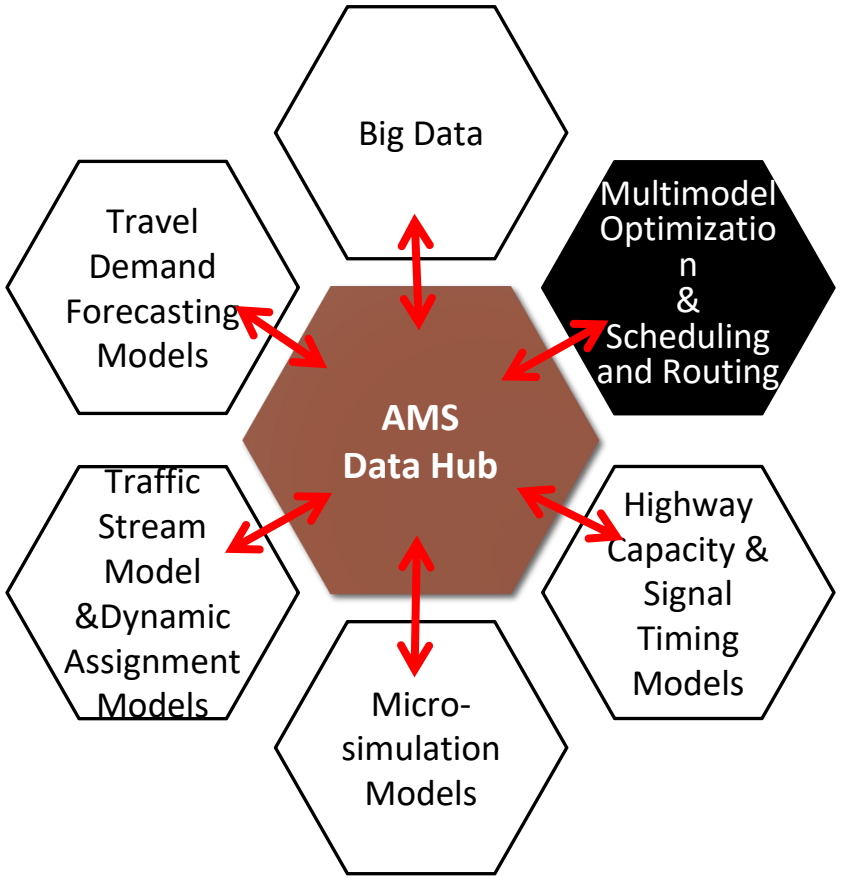
Size of a circle represents the total delay at one node

Color of a circle represents the age of congestion (to identify the congestion propagation sequence)

Statewide Network Coverage in Google Earth



Connection 6: Optimization, Routing and Scheduling

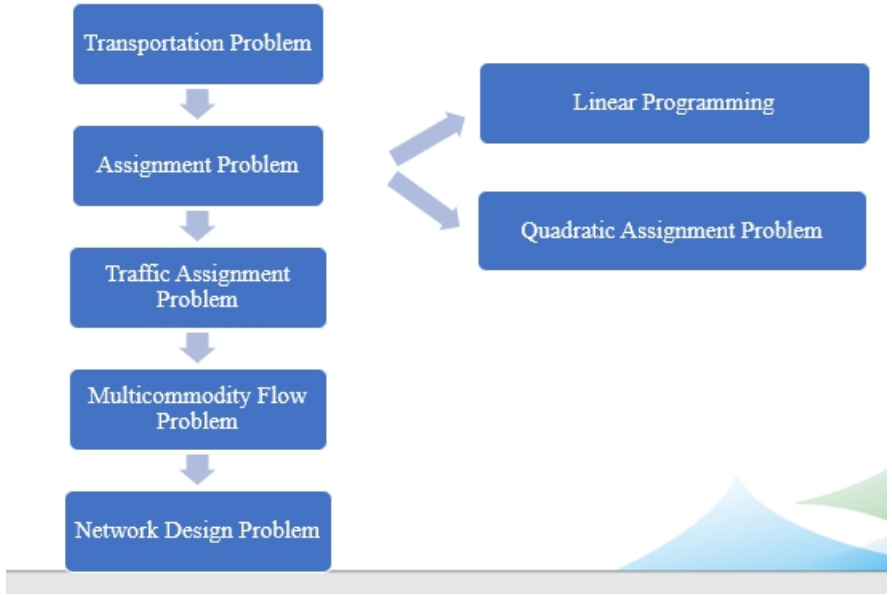


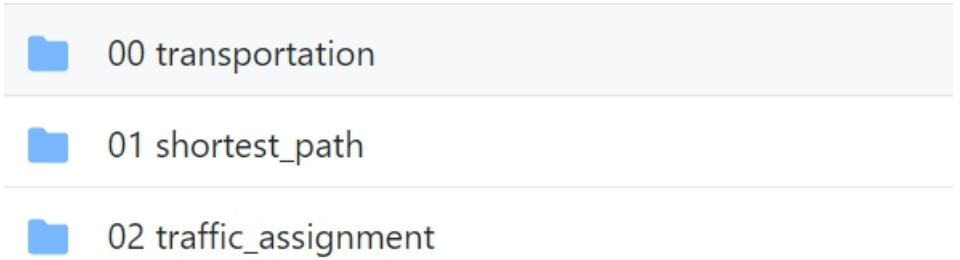
- Optimization
  - Transportation problem
  - Shortest path problem
  - Space time network design
  - Transit network design
  - Yard management
  - Vehicle routing problem
  - Train scheduling
  - Customized bus service design
- https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

- VRPLite
  https://github.com/xzhou99/VRPLite
  DTALite-S: for multimodal transportation simulation
  https://github.com/xzhou99/DTALite-S

6.1 Learning Nework Flow Models based on GAMS

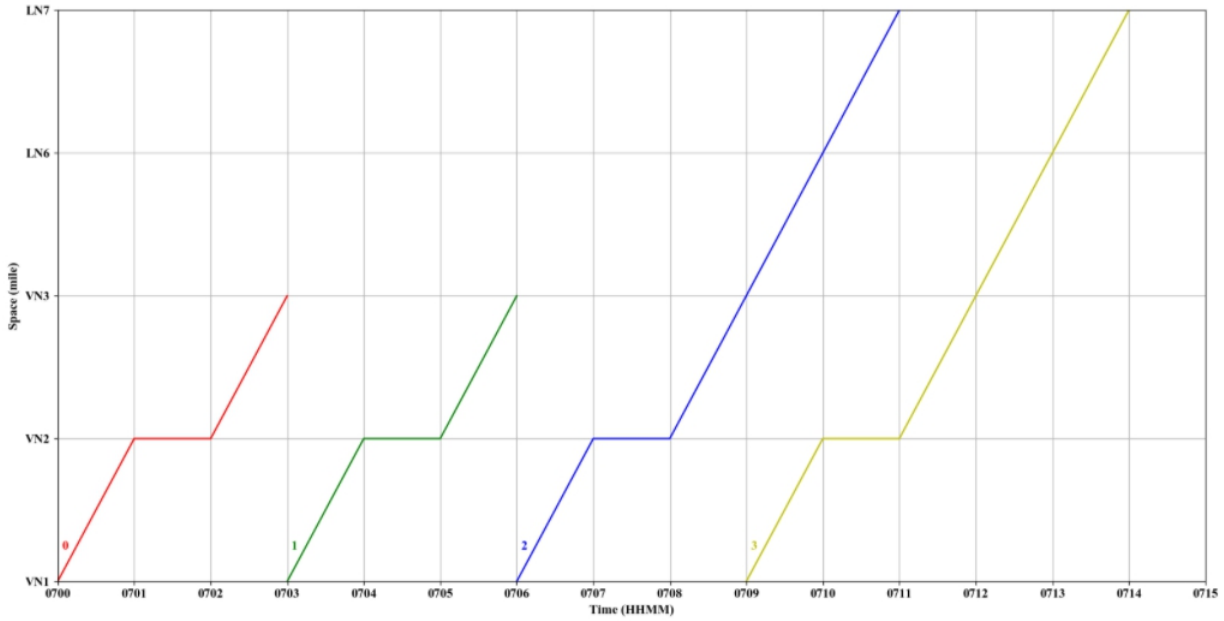
- https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network





6.2 From space-time diagram to space-time networks

https://github.com/xzhou99/space-time-diagram_gmns

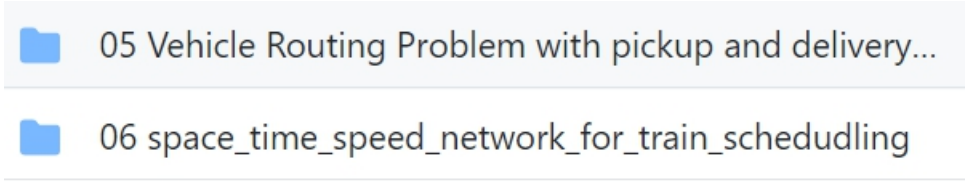


https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

6.3 From space-time network to space-time-state network in vehicle routing

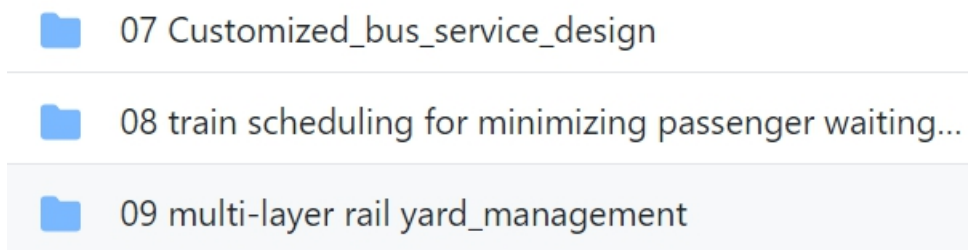
- https://github.com/xzhou99/VRPLite
- https://link.springer.com/article/10.1007/s40864-018-0083-7

https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network



6.4 Use space-time network in multimodal system scheduling

https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network



6.5 Automated vehicle scheduling based on space-time networks

https://github.com/xzhou99/AVRLite

10 multi-vehicle longitudinal trajectory optimization

https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

Objective function:

$$\min Z = \sum_{a} \sum_{(i,j,t,s)} x^{a}_{i,j,t,s} \times c^{a}_{i,j,t,s}$$

Subject to:

Flow balance constraint:

$$\sum_{i,t:(i,j,t,s)\in E} x^{a}_{i,j,t,s} - \sum_{i,t:(j,i,s,t)\in E} x^{a}_{j,i,s,t} = \begin{cases} -1 & j = o(a), s = DT^{a} \\ 1 & j = d(a), s = T \\ 0 & otherwise \end{cases}$$

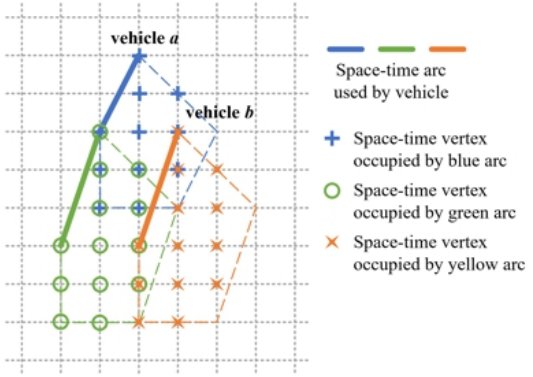
Indicator of vertex visited by vehicles:

$$\sum_{(i,j,t,s)\in\varphi(k,p)} x^{a}_{i,j,t,s} \le y^{a}_{(k,p)} \times M, \qquad \forall a, \forall (k,p)$$

Simplified car-following safety constraints:

$$\sum_{a} y^{a}_{(k,p)} \le 1, \qquad \forall (k,p)$$

Binary variables: $x^{a}_{i,j,t,s} \in \{0,1\}$, $y^{a}_{(k,p)} \in \{0,1\}$



Occupied vertices of the leading and following vehicle

6.6 Travel demand and activity modeling based on space-time-state network

https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

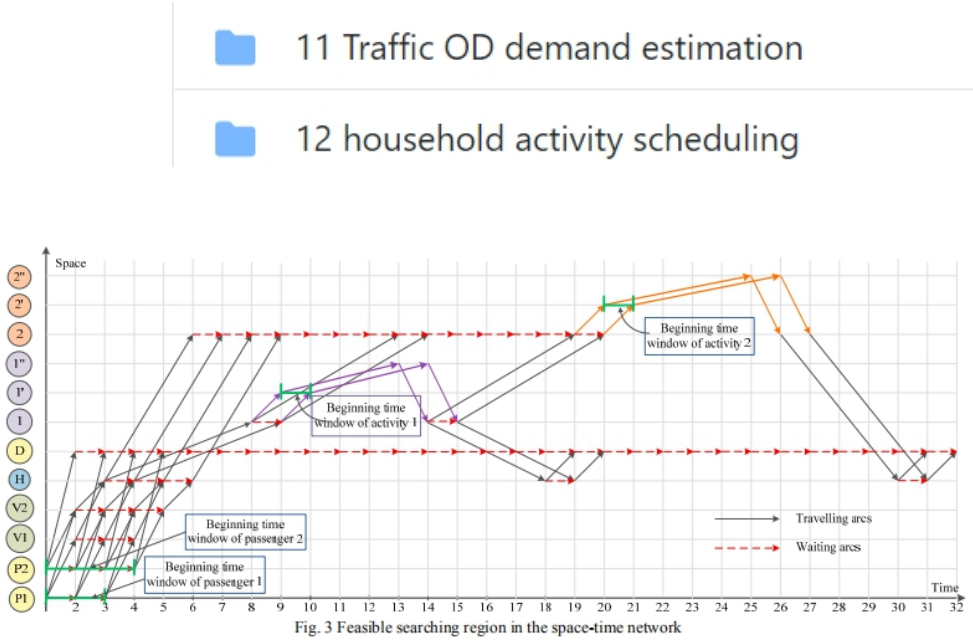


Fig. 3 Feasible searching region in the space-time network

Objective function

$$\min \sum_{v} \sum_{(i,j,t,s,w,w') \in E} (c^{v}_{i,j,t,s,w,w'} \times x^{v}_{i,j,t,s,w,w'})$$

Subject to,

(1) Flow balance constraint for each vehicle:

$$\sum_{i,t,w:(i,j,t,s,w,w') \in E} x^{v}_{i,j,t,s,w,w'} - \sum_{i,t,w:(j,i,s,t,w',w) \in E} x^{v}_{i,j,t,s,w,w'} = \begin{cases} -1 & j = O(v), s = DT(v), w = [0,0,\ldots,0] \\ 1 & j = D(v), s = T, w = [2,\ldots,2] \\ 0 & otherwise \end{cases}, \forall v$$

(2) Mandatory activity performing constraint for the driver on the activity arcs (including ride-sharing):

$$\sum_{i,t,w:(i,j,t,s,w,w') \in E(v,a_m)} x^{v}_{i,j,t,s,w,w'} = 1, \ \forall v, \forall a \in A(v)$$

(3) Binary variable: $x^{v}_{i,j,t,s,w,w'} = \{0,1\}$

Summary of related open-source tools

Simulation-based mesoscopic dynamic traffic assignment

DTALite(based on simplified kinematic wave model)

Simulation-based mesoscopic dynamic traffic assignment

STALite

visualization interfeace

NEXTA

Traffic state estimation and prediction, Train routing and scheduling

VRPLite