Learning Transportation Modeling using Case Studies, GMNS standard, Open-Source AMS Tools

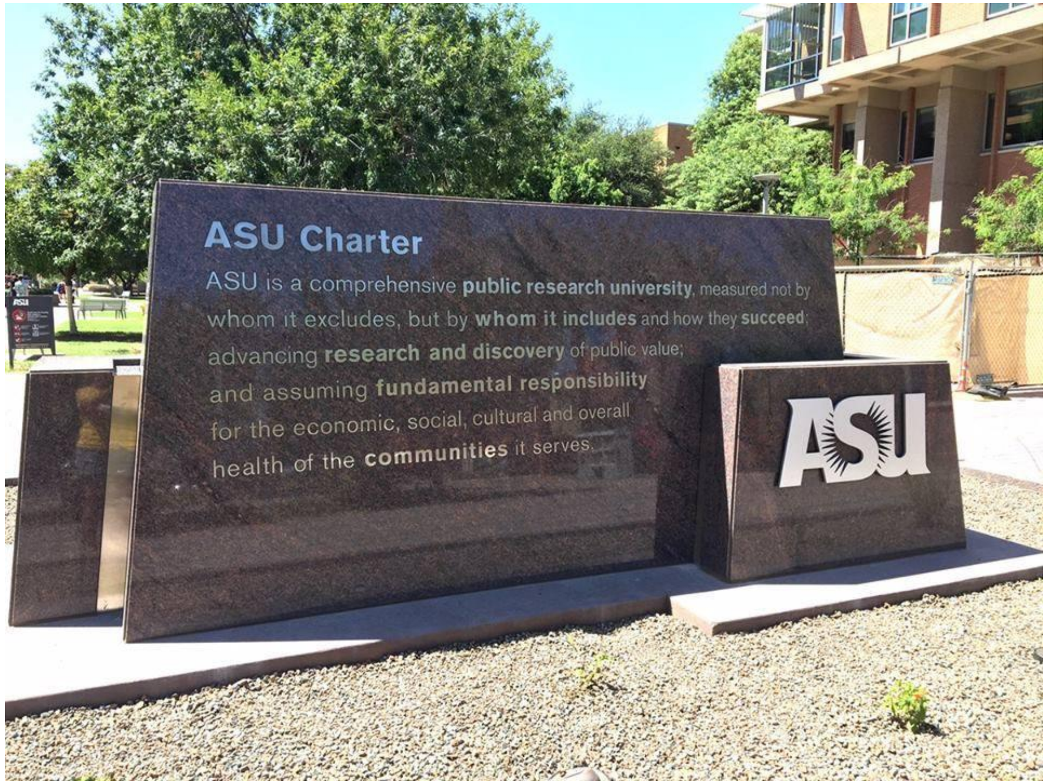
Developed through an early FHWA project: Effective Integration of Analysis, Modeling, and Simulation Tools
AMS DATA HUB CONCEPT OF OPERATIONS:
https://www.fhwa.dot.gov/publications/research/operations/13036/004.cfm

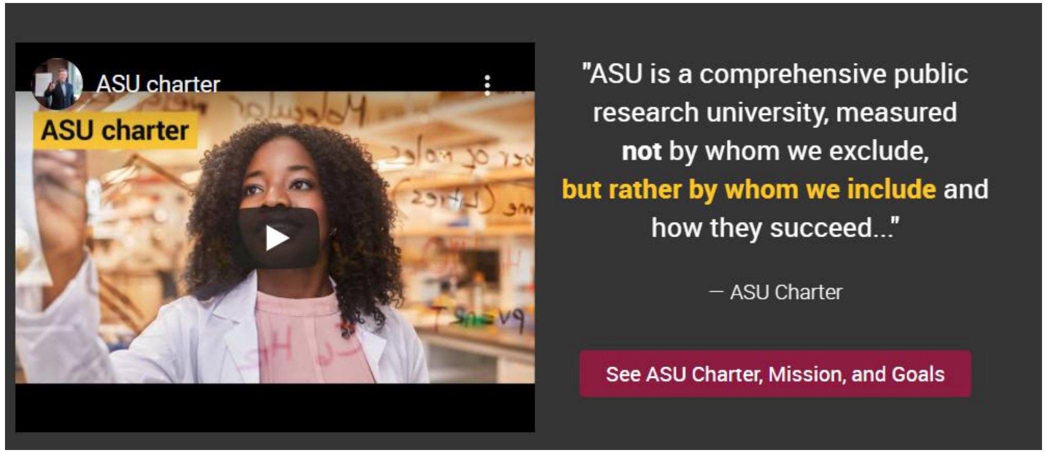
General Modeling Network Specification (GMNS): https://github.com/zephyr-data-specs/GMNS

Xuesong (Simon) Zhou (xzhou74@asu.edu)
https://github.com/xzhou99

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.

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

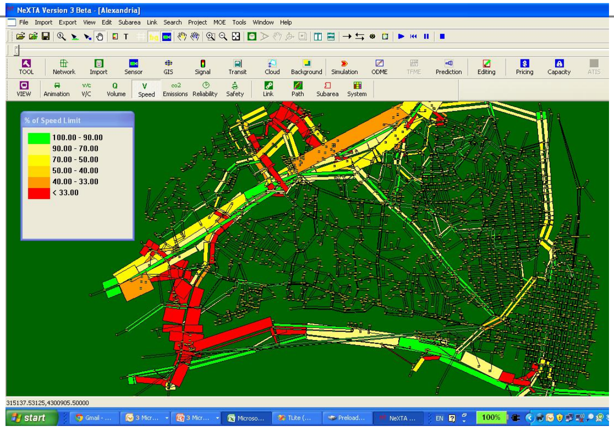
- Simplified data input from static traffic assignment
- Use parallel computing capability, simplified routing and simulation
- Open-source

Visualization: Seeing is believing
Excel Tools: Start from basics

Open-source Free Software Package

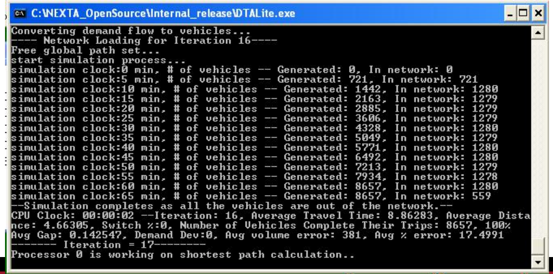
NEXTA: front-end Graphical User Interface GUI (C++)

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



DTALite: Open-source computational engine (C++)

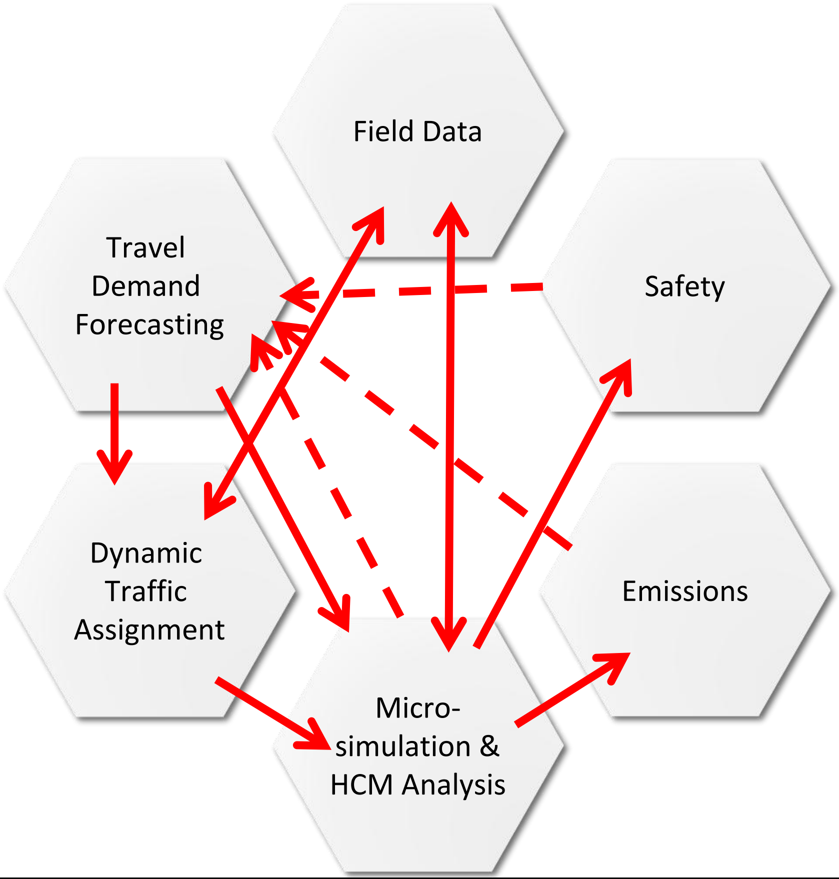
- STALite: Simulation-based mesoscopic dynamic traffic assignment
- https://github.com/xzhou99/stalite-dtalite_software_release
    - Light-weight and agent-based DTA
    - Simplified kinematic wave model (Newell)
    - Built-in OD demand matrix estimation (ODME) program
    - Emission prediction (light-weight MOVES interface)
        - Simplified car follow modeling (Newell)



2. Integrated Modeling Practices

Current Practice

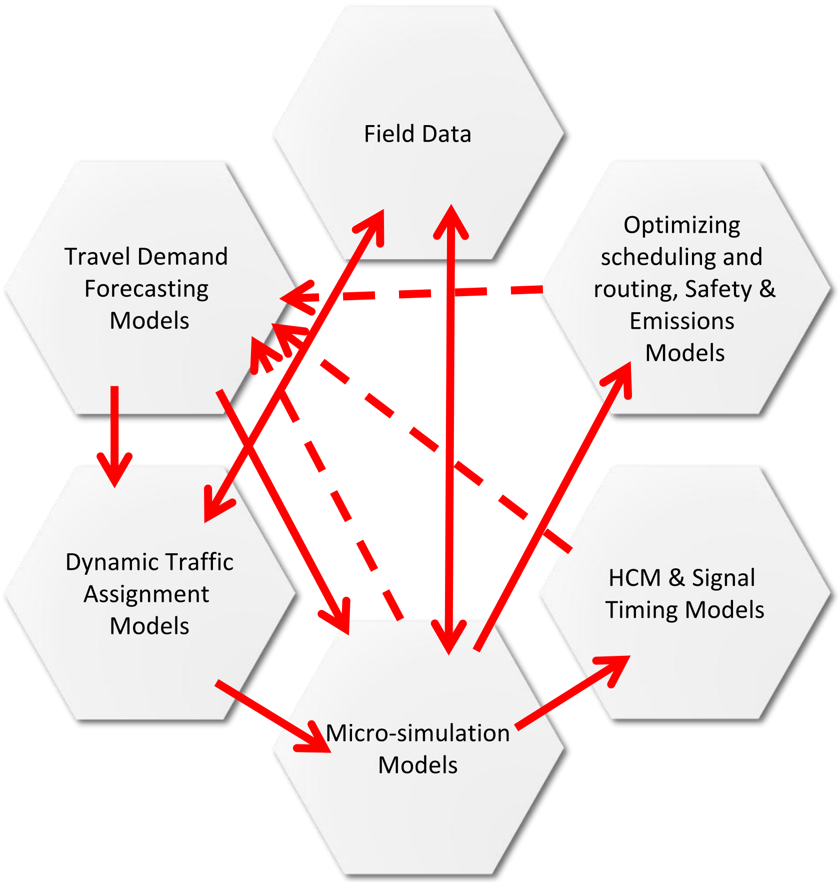
Ad Hoc



Our Vision

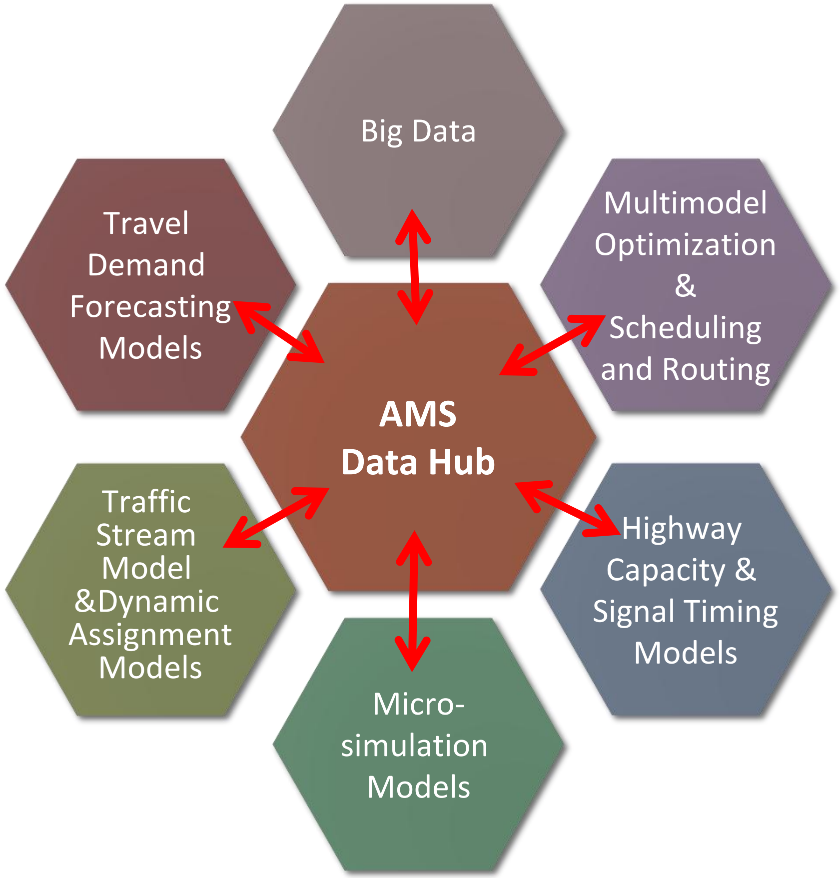
Current Practice

Ad Hoc



With Analysis, Modeling and Simulation (AMS) Data Hub

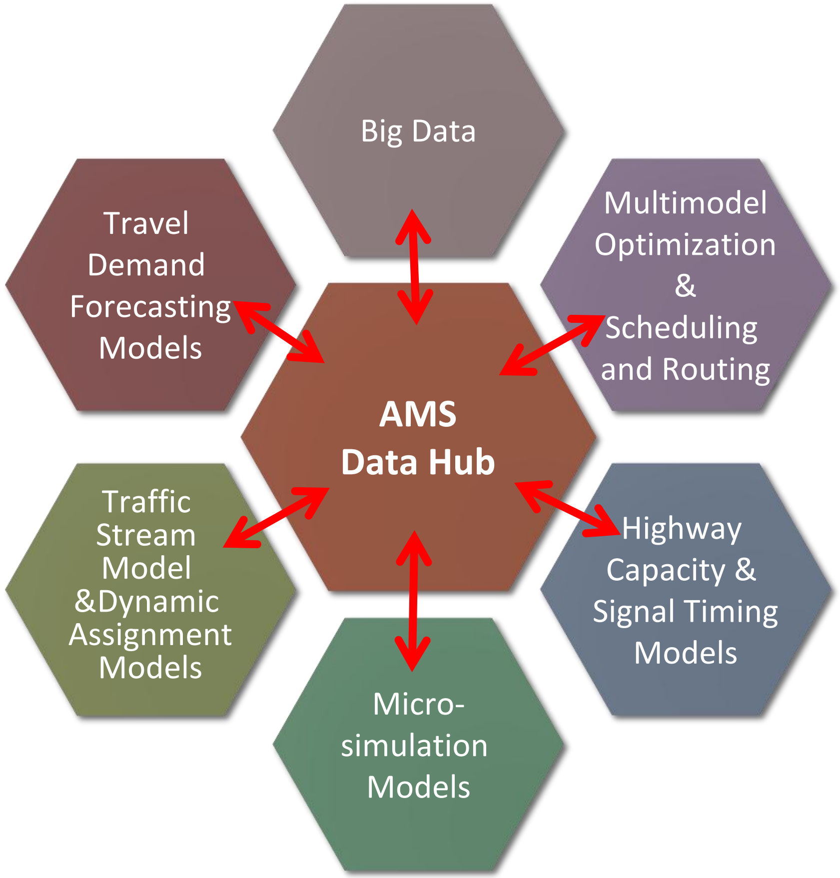
Systematic



Our Vision

With AMS Data Hub

Systematic

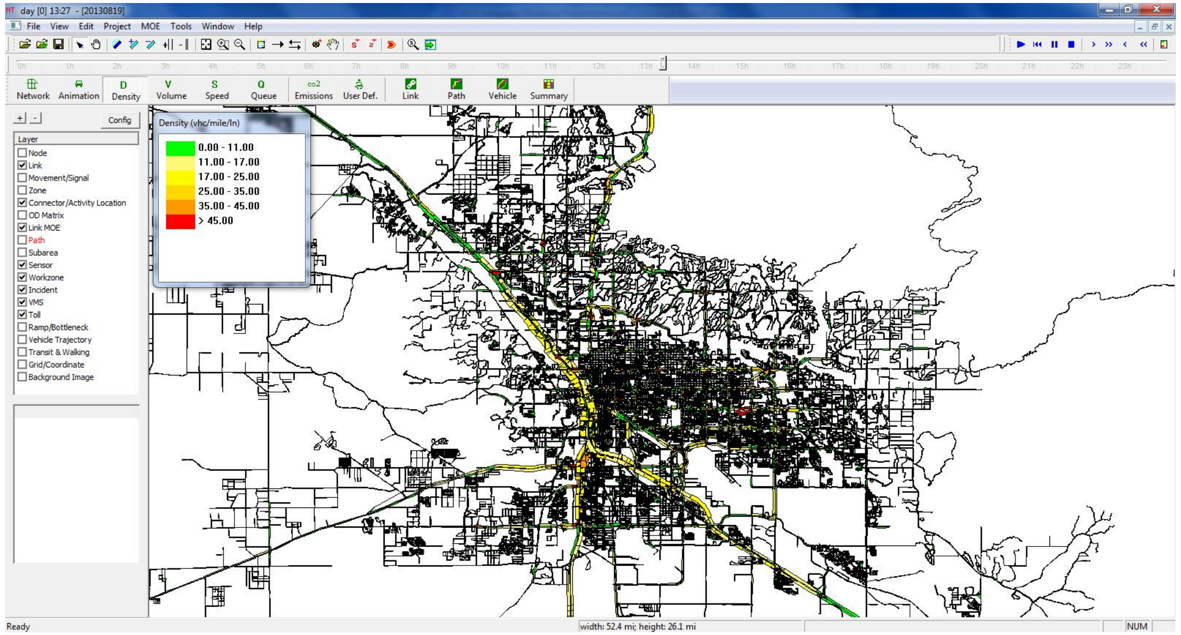


3. AMS Data Hub Software Prototype: NeXTA

Start with GIS with multiple transportation layers

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

Focus on data processing and visualization



Sample data set: Pasadena Test Data Sets

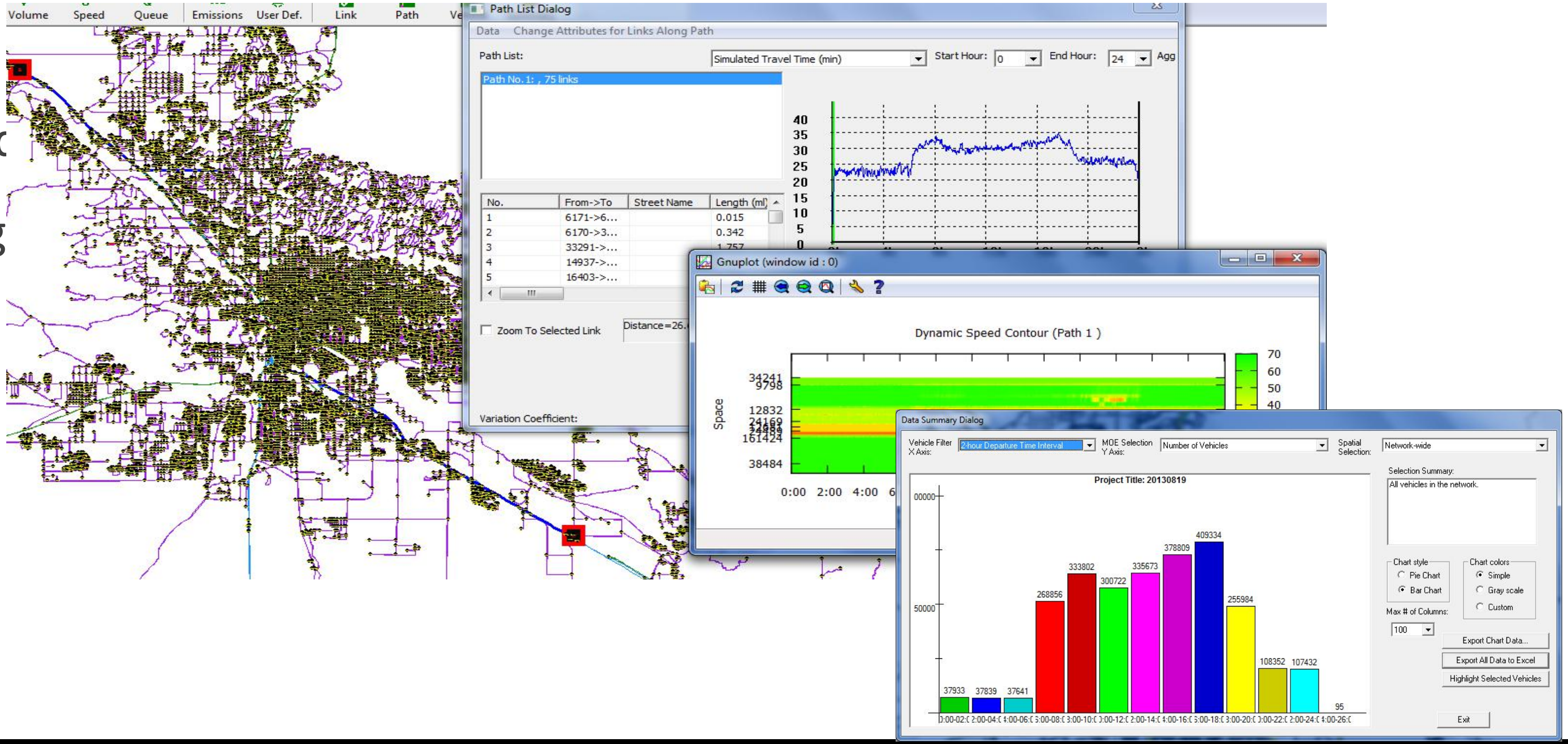
https://its.dot.gov/data/

https://datahub.transportation.gov/Automobiles/Pasadena-Test-Data-Sets/xig6-cb63

24-hour Simulation Results

2.6 Million Vehicles, 6 GB of d

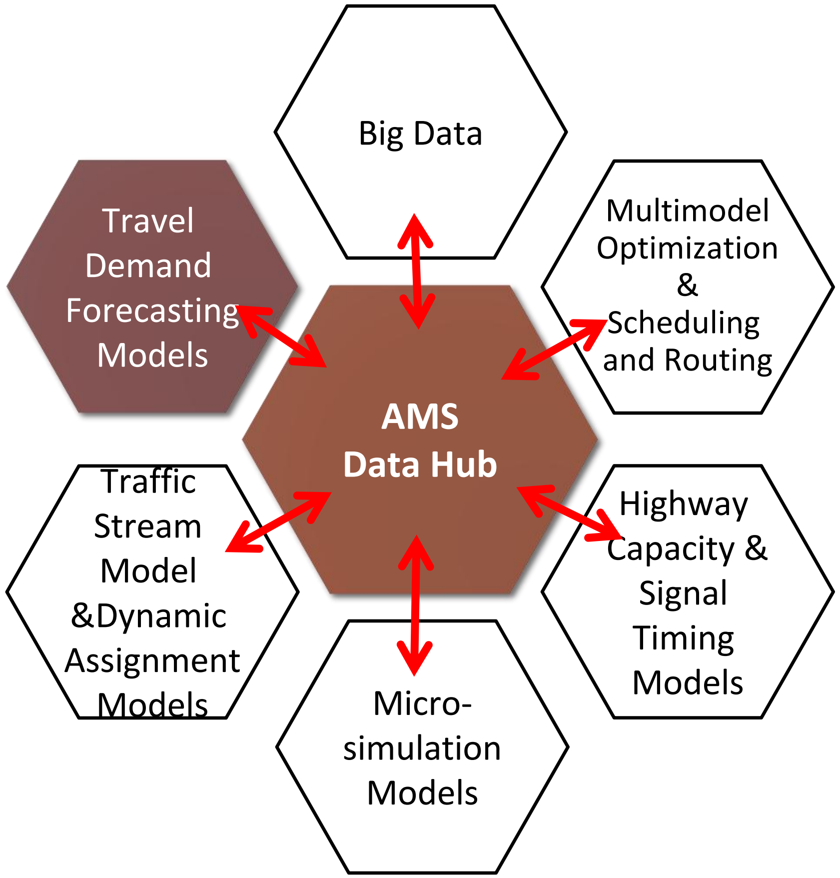
Users can select any path to g



Sample data sets

- 00_six_node
- 01_two_corridor
- 02_Braess's_paradox
- 03_three_corridor
- 04_Sioux_Falls
- 05_Chicago_sketch
- 06_Chicago_regional
- 07_West_Jordan_Utah
- 08_Tempe ASU Network
- 09_NGSIM_trajectory
- 10_Arizona_ICM
- 11_multi_modal_simulation_DTALite-S...
- 12_Berkeley_Highway_Lab-Network

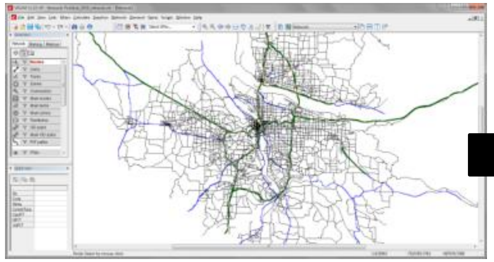
Data hub 0: Network



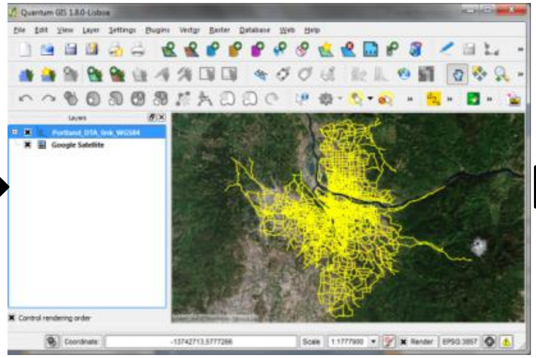
- Based on GMNS data format
  - node, link, movement
  - https://github.com/zephyr-data-specs/GMNS
- Regional planning models through GIS shape files
  - TransCAD
  - CUBE
  - VISUM
- Export network to GIS
- Subarea cut utility

Approach 1: Network Data GIS shape file Importing

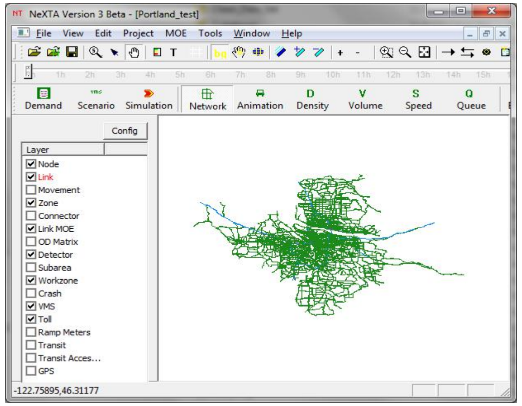
- Import GIS node/link/zone shape files
  - Utilize Open-source Geospatial Data Abstraction Library (GDAL) library
- https://github.com/xzhou99/NeXTA-GMNS/tree/master/tools/GIS_shape_file_CSV_file_conversion



Cube/TransCAD/VISUM

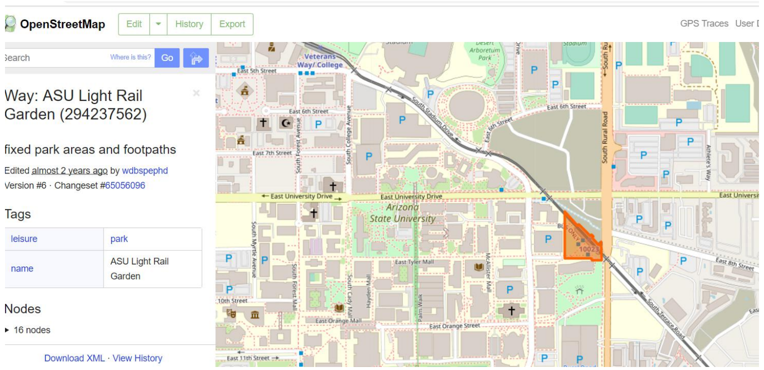


Open-Source Q-GIS



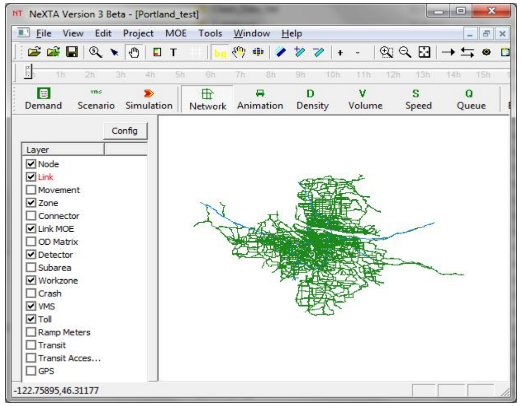
Approach 2: Openstreet Map Data Importing

- Import node/link/zone shape to GMNS https://github.com/jiawei92/OSM2GMNS



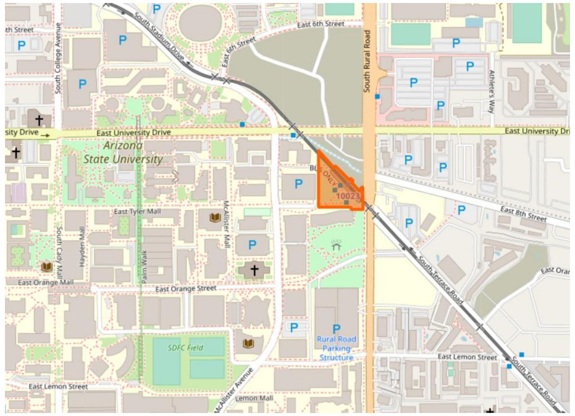
Openstreet Map

OSM2GMNS tool



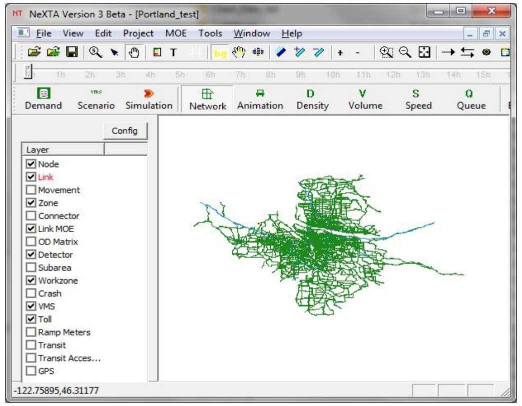
Approach 3: Create network from the scratch based on background image

- Import node/link to GMNS data format



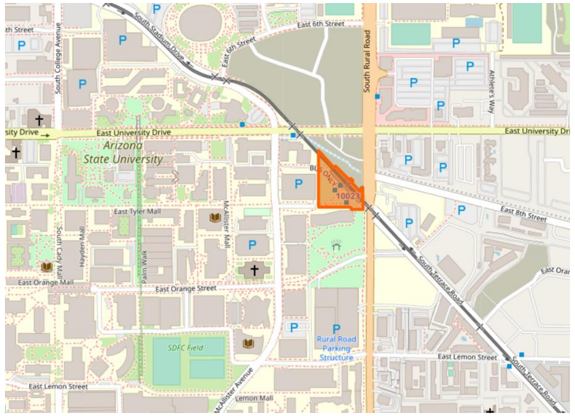
background image as image.bmp

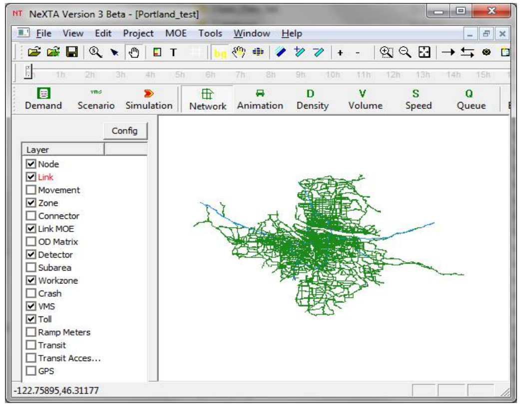
draw nodes and links



Approach 4: General Transit Feed Specification (GTFS)

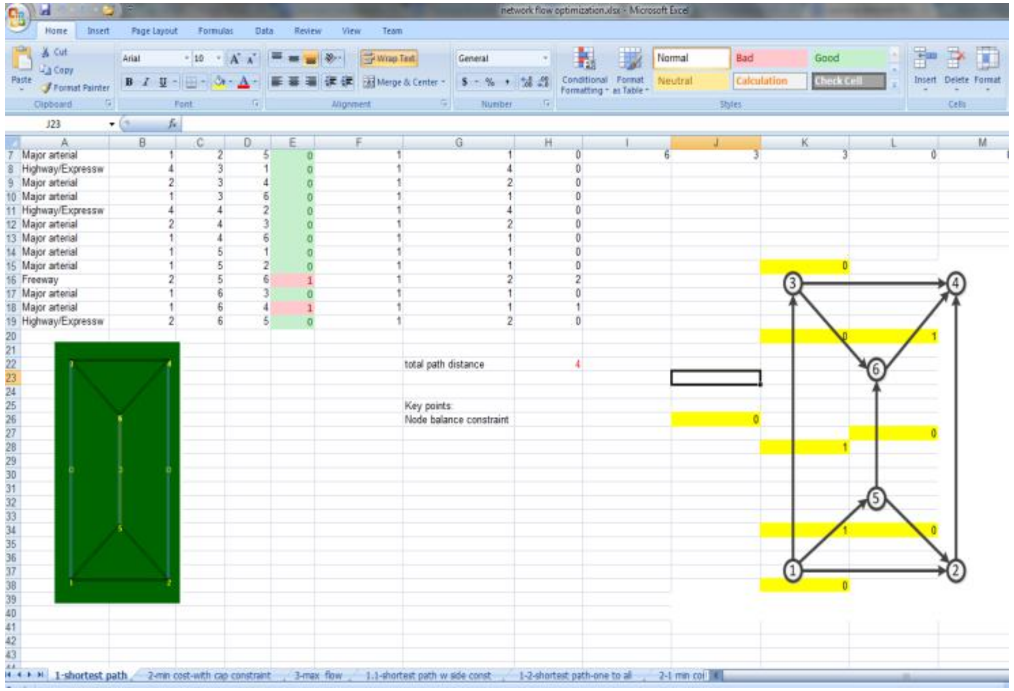
- Import GTFS to node/link data to GMNS data format (NeXTA-> menu -> import GTFS)





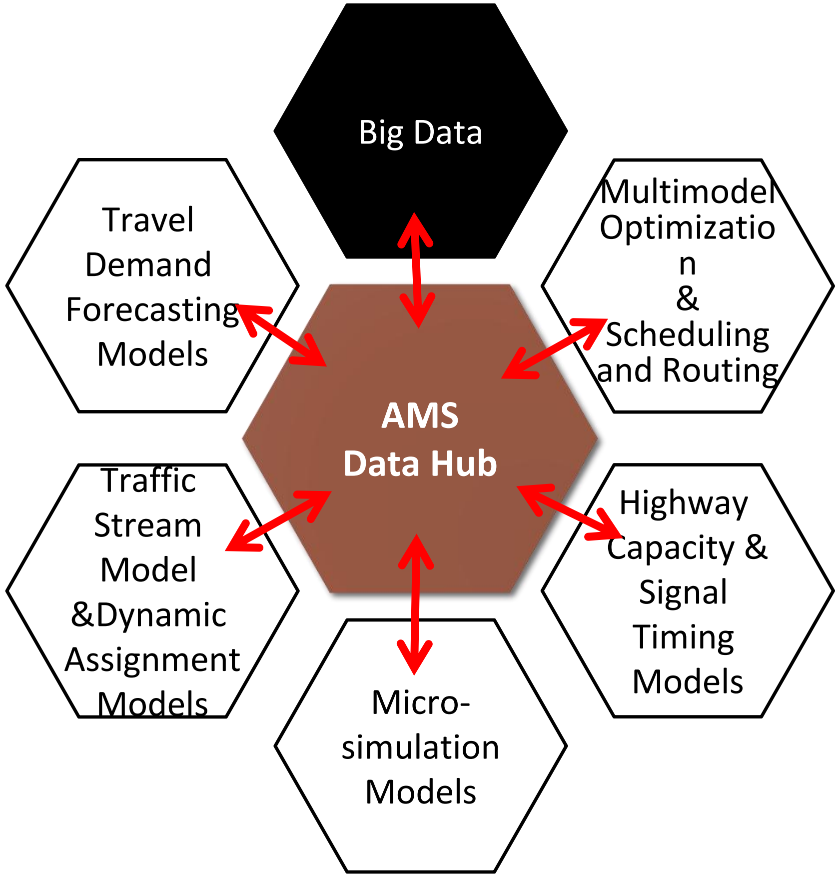
background image as image.bmp draw nodes and links

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

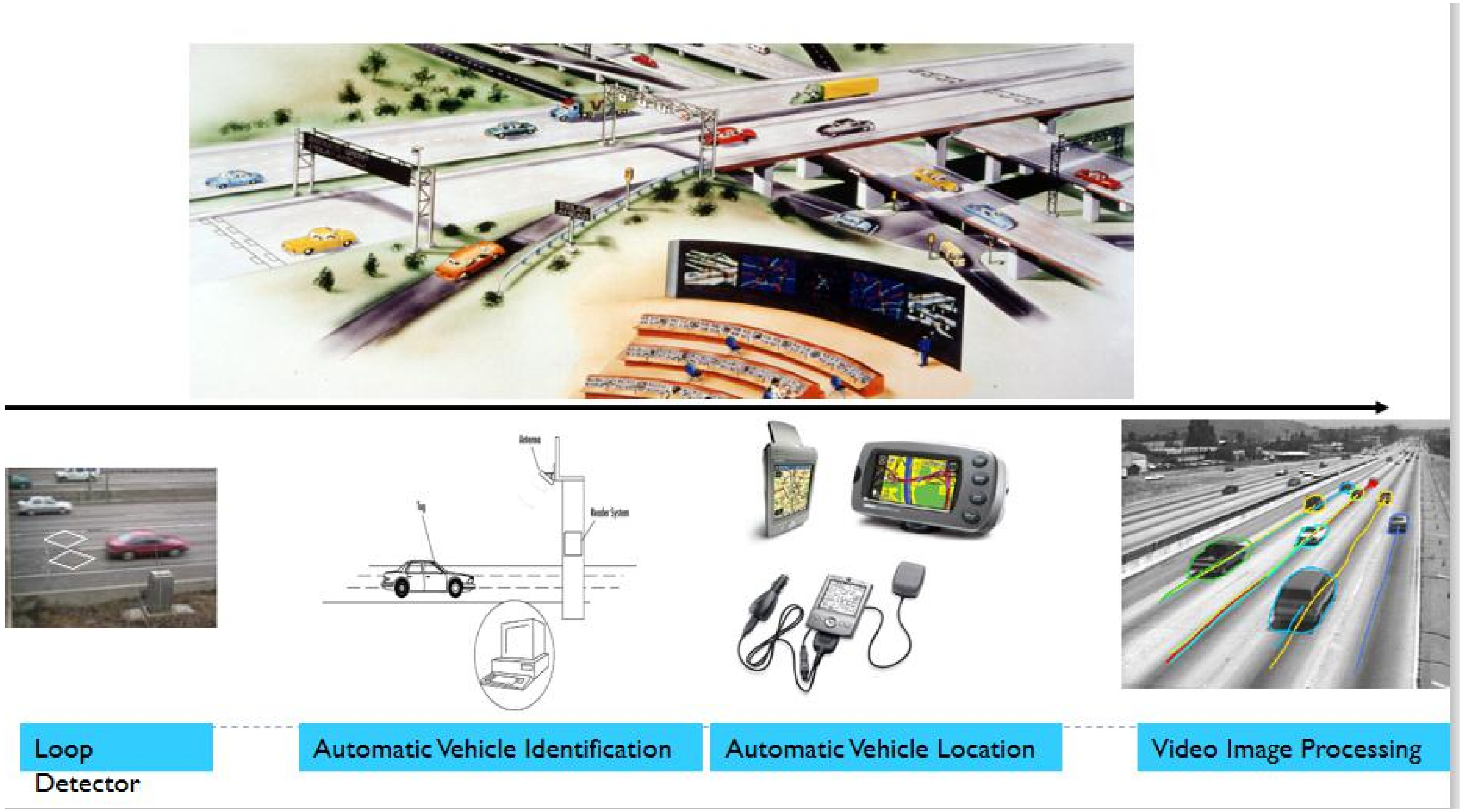


- Sensor data (link_performance.csv)
  15-min speed or count
- GPS trajectory data (trajectory.csv)
- NGSIM trajectory data set
  https://github.com/xzhou99/NeXTA_4_NGSIM_Trajectory_Visualization

BHL sensor data set

I-405 data set from PeMS (to do)

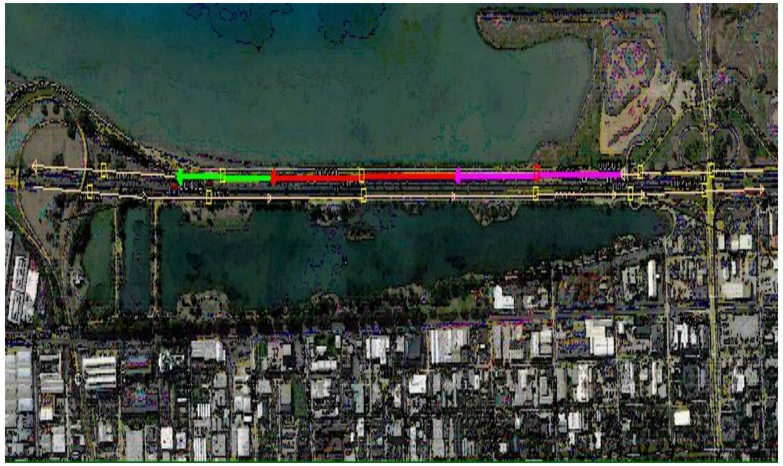
Big data for integrated traffic management

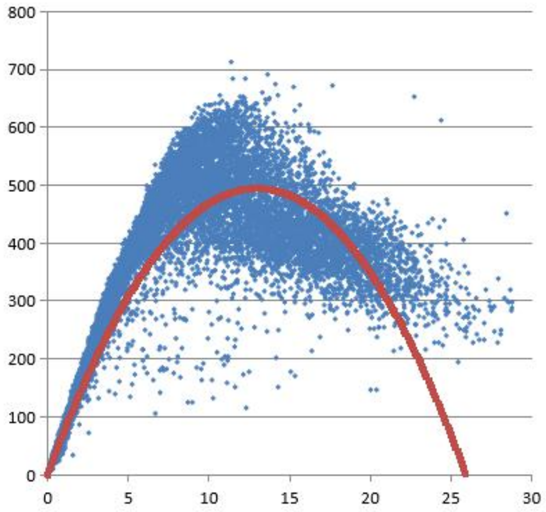


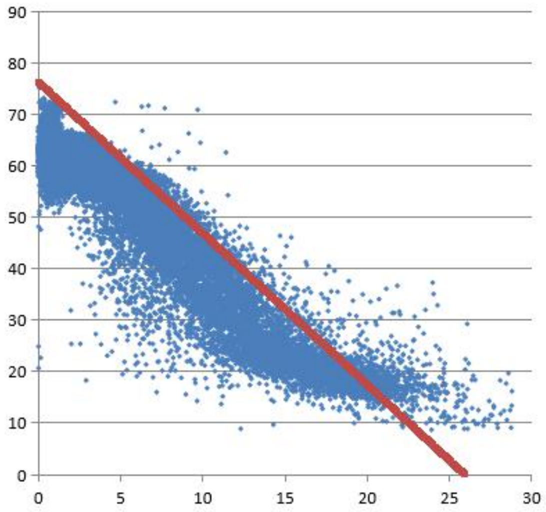
1.1 Understand QKV relationship from aggregated sensor data

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

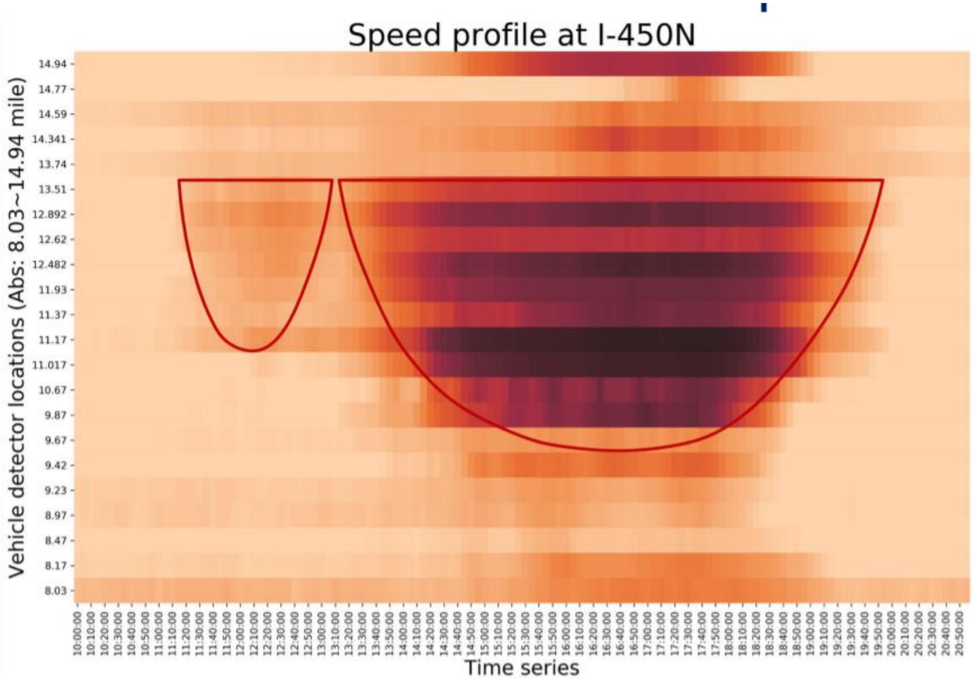


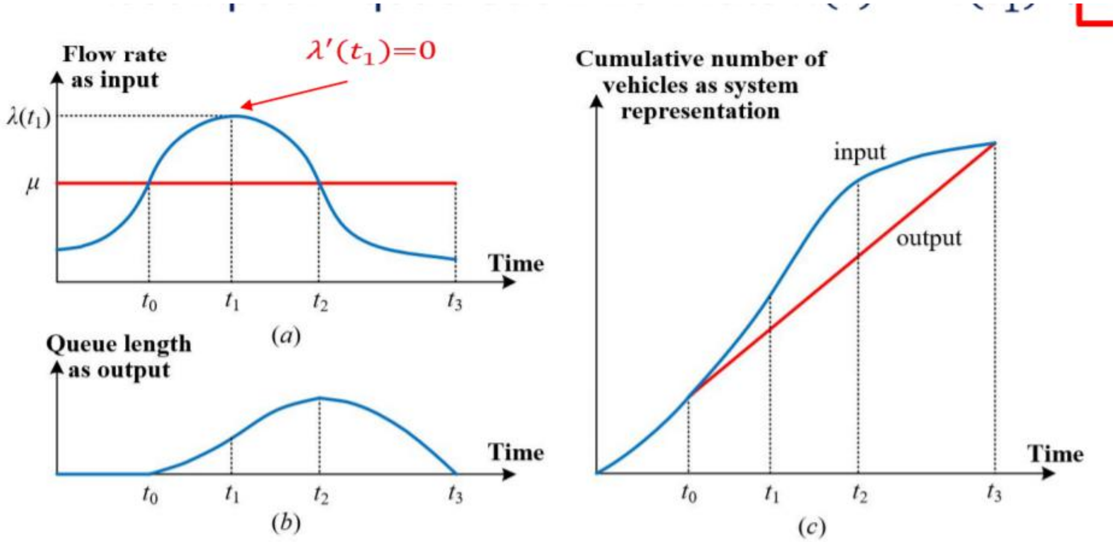




1.2 Understand capacity and bottleneck

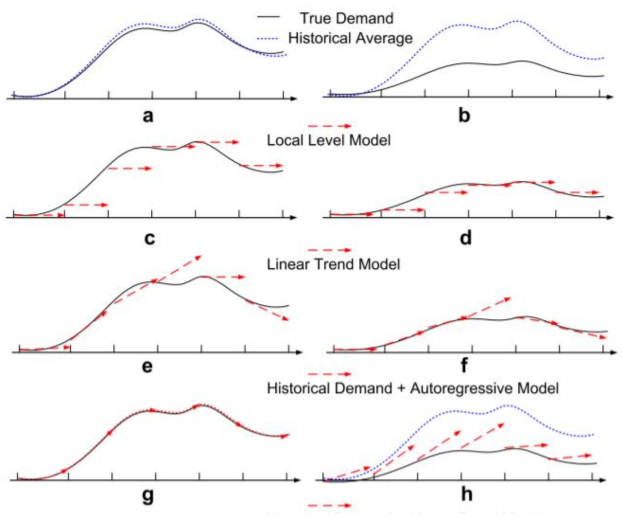
https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/4_highway_capacity_traffic_stream_model

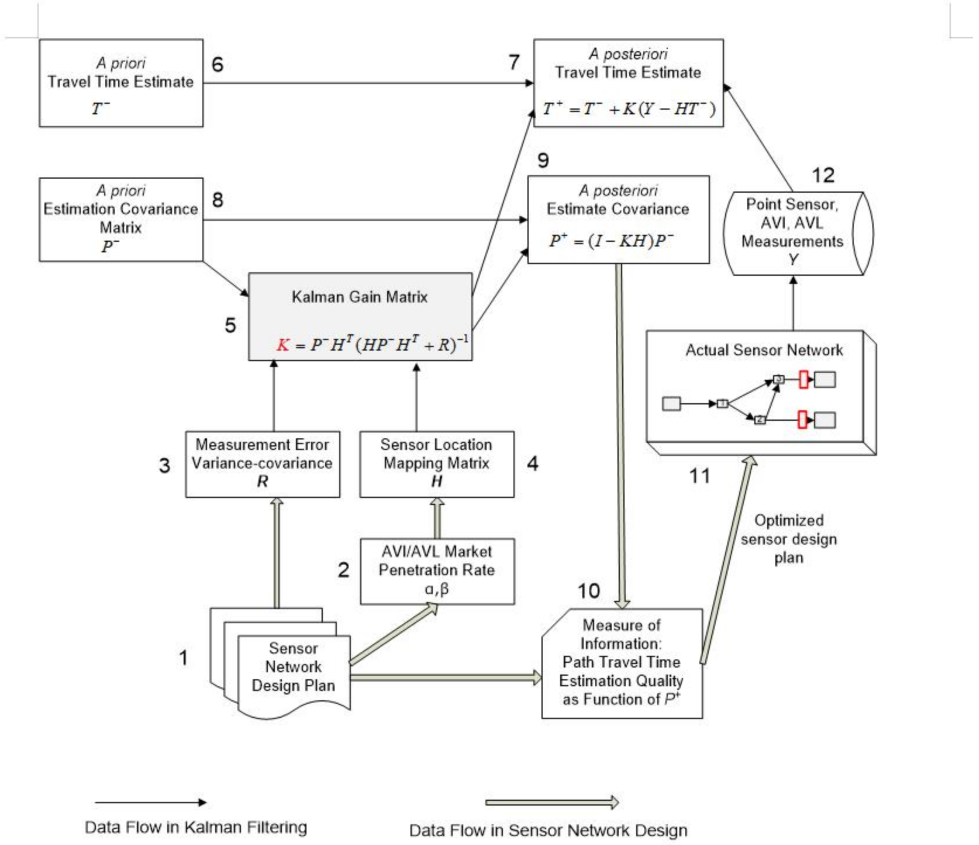




1.3 Understand traffic state estimation and prediction

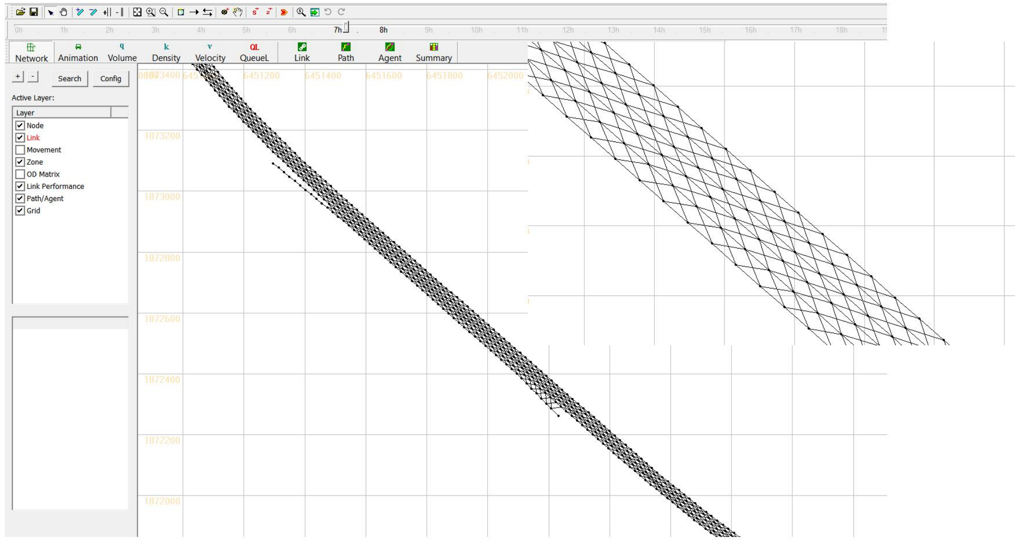
https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/4_highway_capacity_traffic_stream_model/Learning_KalmanFiltering_for_travel_state_estimation_prediction

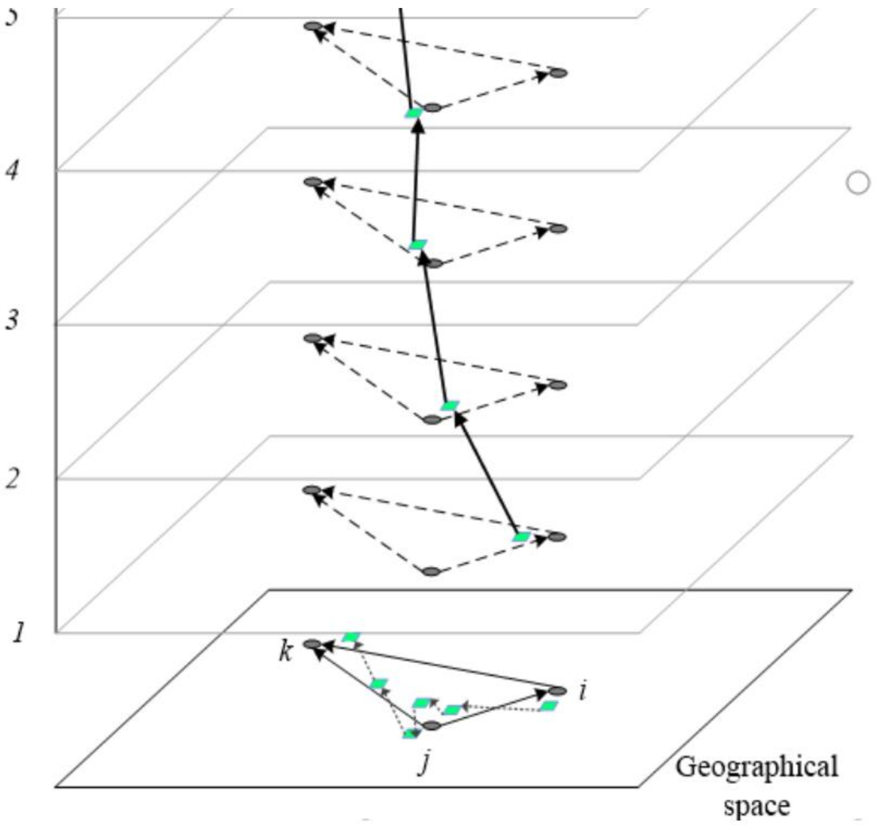




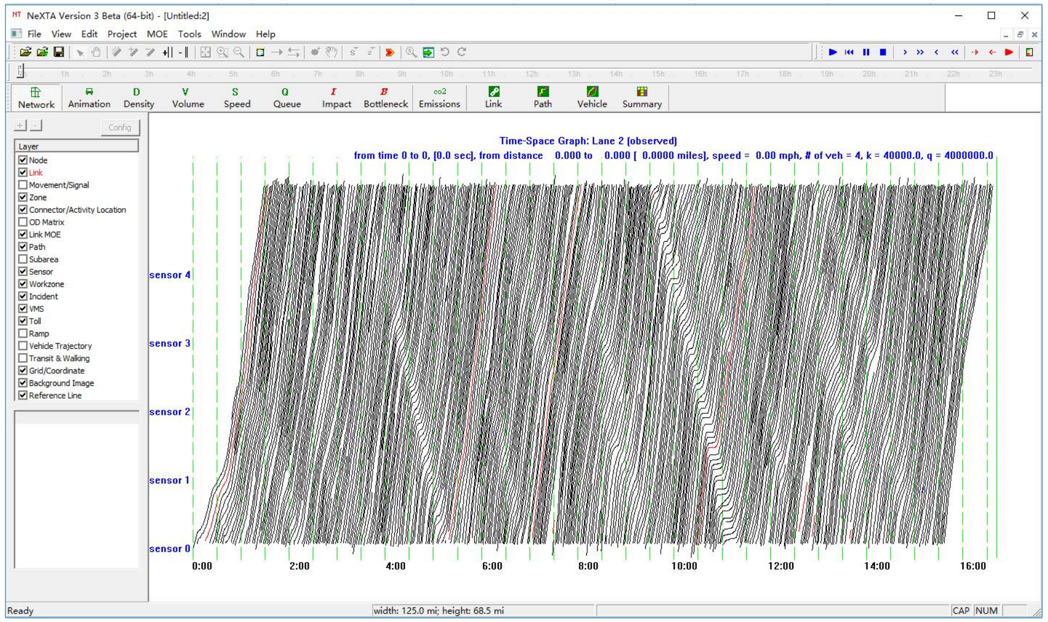
1.4 Map matching based on space-time paths and a time geographic method

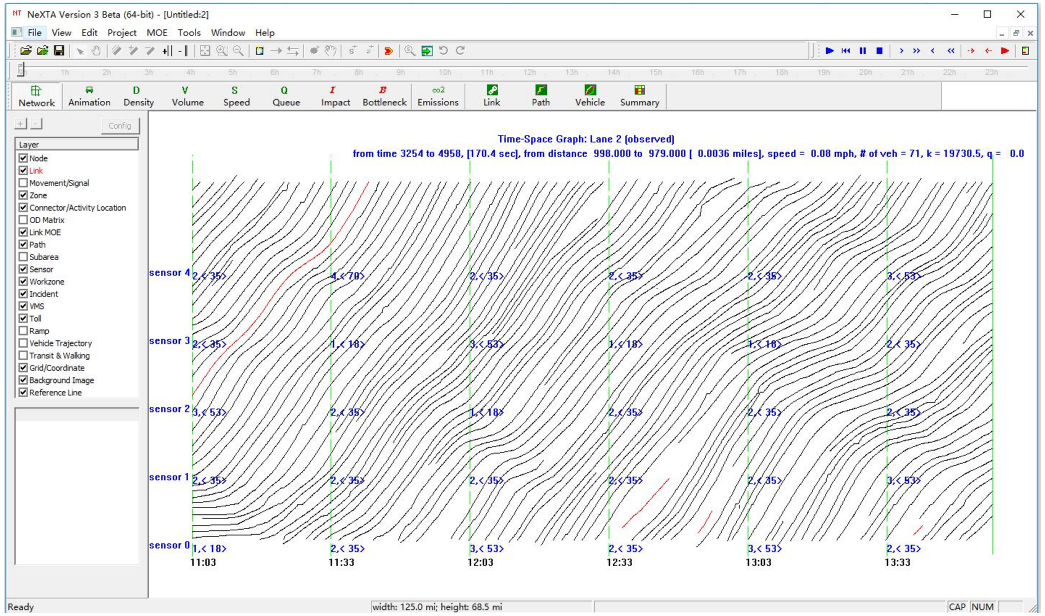
https://github.com/xzhou99/space-time-network-based-map-matching





1.5 Analysis of Trajectories from NGSIM

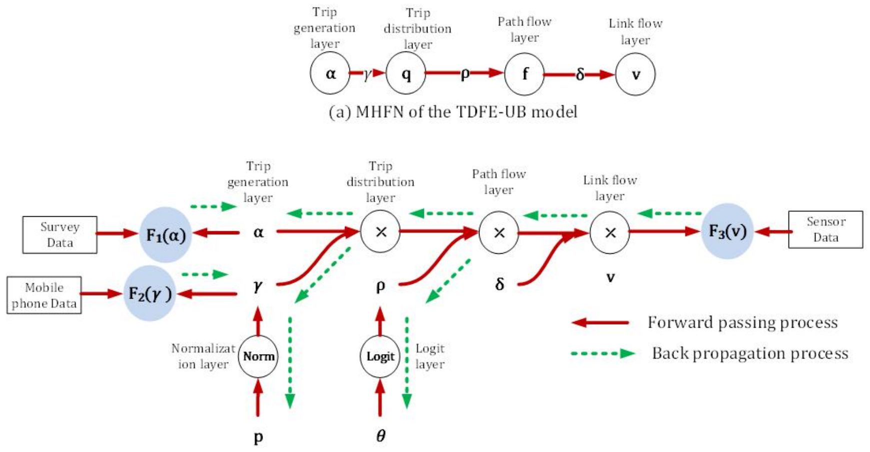




1.6 Integrate deep learning methods with transportation modeling

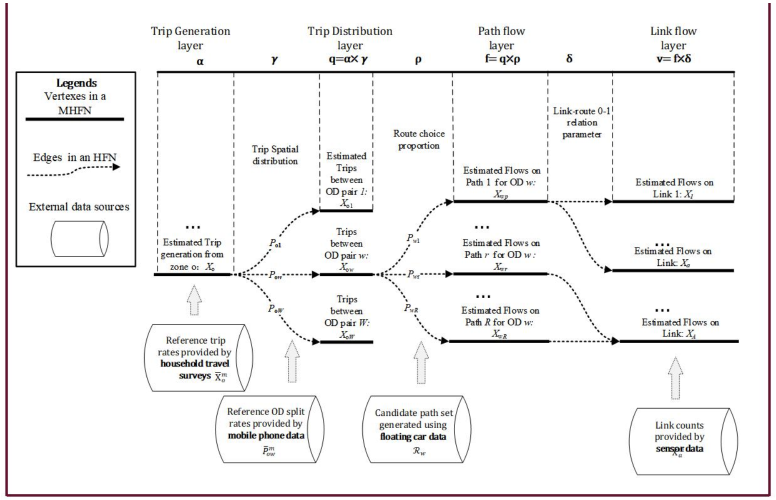
https://github.com/xzhou99/TCGLite

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

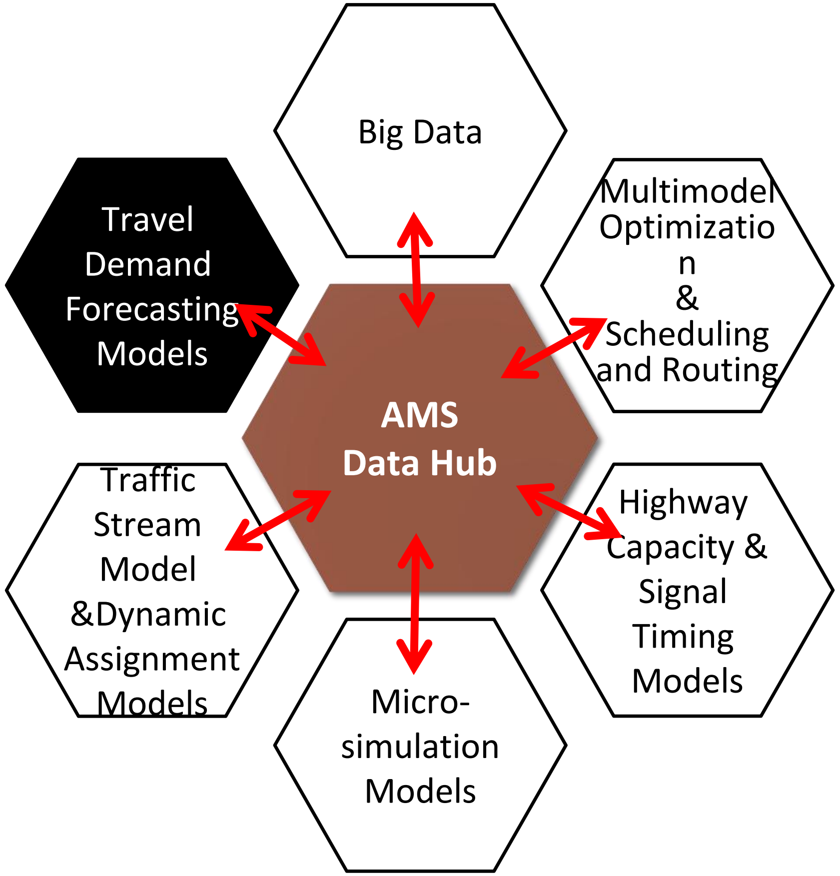


1. **The following flow conservation constraints are expressed by the "Neural Network" whose activation functions are ReLu function** $f(x) = \max(0, x)$:

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

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

Connection 2: Travel demand model

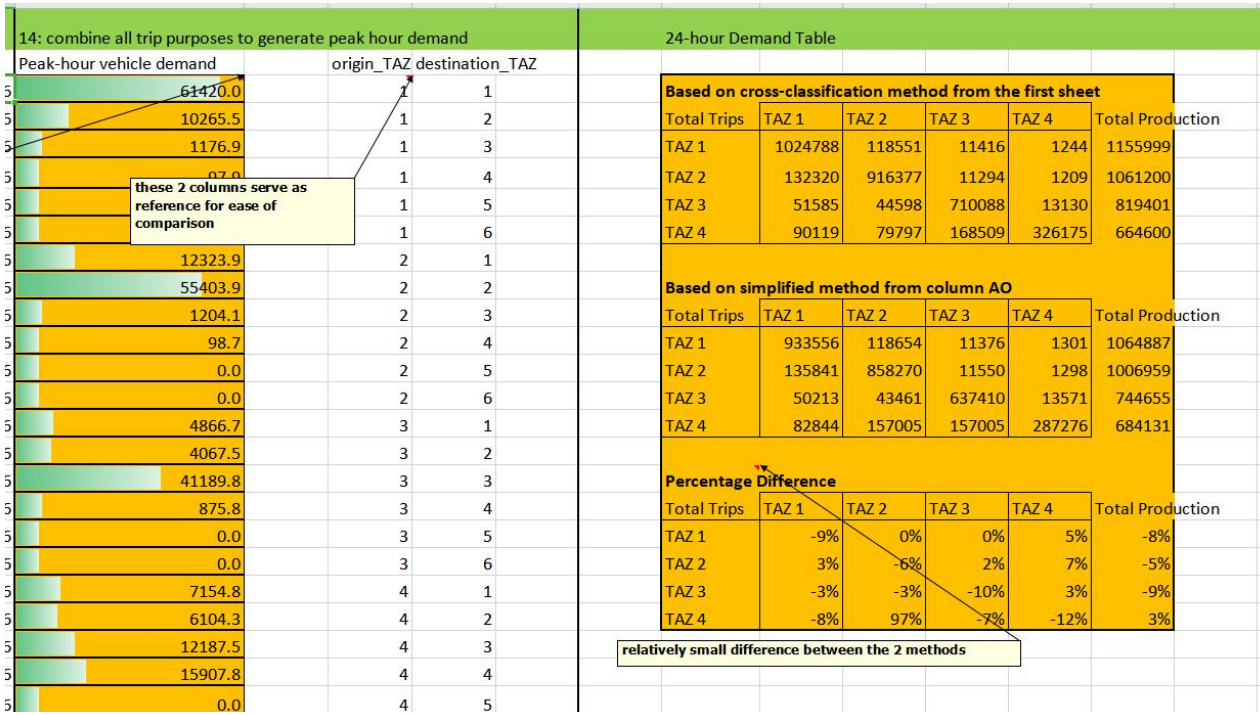


- Trip generation, distribution and model choice
- Static Traffic Assignment
- OD destimation using multiple data sources
- Layout and land use

2.1 Trip generation, distribution and model choice

- Illustration of 4-steps through Excel Spreadsheet
  - https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/6_trip_generation

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



14: combine all trip purposes to generate peak hour demand

Peak-hour vehicle demand	origin_TAZ	destination_TAZ
61420.0	1	1
10265.5	1	2
1176.9	1	3
97.0	1	4
[illegible]	1	5
[illegible]	1	6
12323.9	2	1
55403.9	2	2
1204.1	2	3
98.7	2	4
0.0	2	5
0.0	2	6
4866.7	3	1
4067.5	3	2
41189.8	3	3
875.8	3	4
0.0	3	5
0.0	3	6
7154.8	4	1
6104.3	4	2
12187.5	4	3
15907.8	4	4
0.0	4	5

these 2 columns serve as reference for ease of comparison

24-hour Demand Table

Based on cross-classification method from the first sheet

Total Trips	TAZ 1	TAZ 2	TAZ 3	TAZ 4	Total Production
TAZ 1	1024788	118551	11416	1244	1155999
TAZ 2	132320	916377	11294	1209	1061200
TAZ 3	51585	44598	710088	13130	819401
TAZ 4	90119	79797	168509	326175	664600

Based on simplified method from column AO

Total Trips	TAZ 1	TAZ 2	TAZ 3	TAZ 4	Total Production
TAZ 1	933556	118654	11376	1301	1064887
TAZ 2	135841	858270	11550	1298	1006959
TAZ 3	50213	43461	637410	13571	744655
TAZ 4	82844	157005	157005	287276	684131

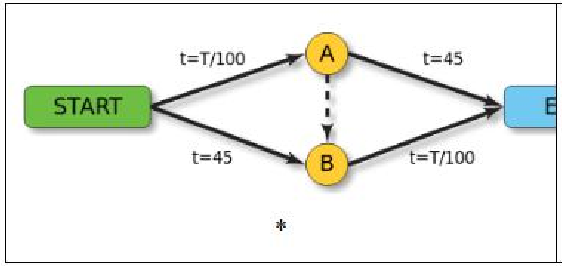
Percentage Difference

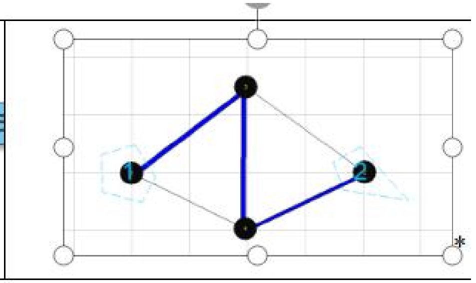
Total Trips	TAZ 1	TAZ 2	TAZ 3	TAZ 4	Total Production
TAZ 1	-9%	0%	0%	5%	-8%
TAZ 2	3%	-6%	2%	7%	-5%
TAZ 3	-3%	-3%	-10%	3%	-9%
TAZ 4	-8%	97%	-7%	-12%	3%

relatively small difference between the 2 methods

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

$$r(w,p)^{m+1} = \mathrm{Max}\{0, r(w,p)^m - \alpha^{(m)}[\beta_r \nabla h^r(r) + \beta_q \nabla h^q(r) + \beta_o \nabla h^o(r) + \beta_d \nabla h^d(r)]\} \quad (3)$$

where $\alpha^{(m)}$ is the step size, and the gradients, which consist of the first-order partial derivatives with respect to a path flow variable $r(w,p)$, can be derived as follows.

$$\nabla h^r(r) = \frac{\partial \left[\sum_{p\in P} r(w,p) - \bar{d}(w)\right]^2}{\partial r(w,p)} = 2\left(\sum_{p\in P} r(w,p) - \bar{d}(w)\right) \quad (4)$$

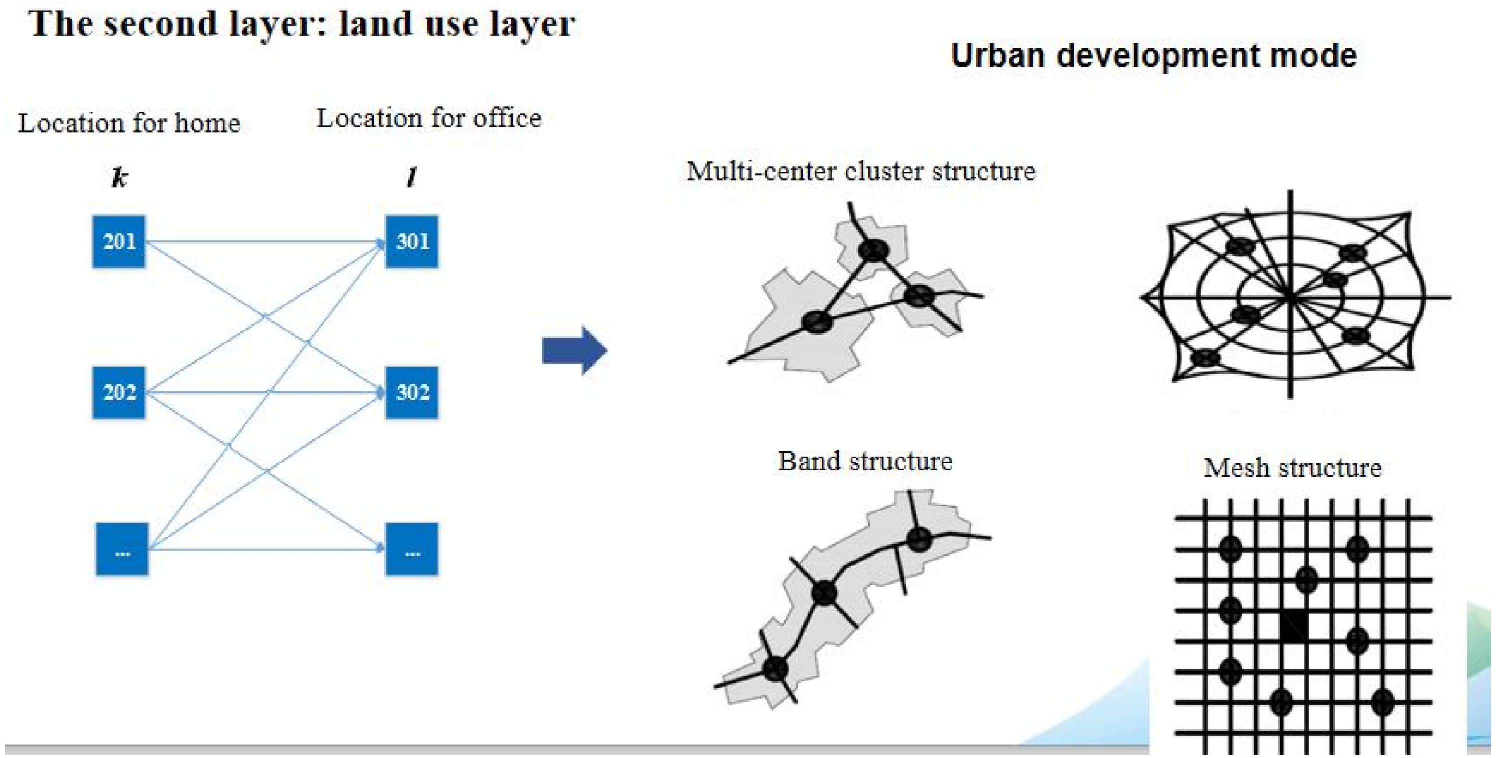
$$\nabla h^q(r) = \frac{\partial \sum_{l\in A} [q(l) - \bar{q}(l)]^2}{\partial r(w,p)} = 2\sum_{l\in A}\left\{[q(l) - \bar{q}(l)] \times \frac{\partial q(l)}{\partial r(w,p)} \times \theta^l_{w,p}\right\} \quad (5)$$

$$\nabla h^o(r) = \frac{\partial \sum_{i\in I} [O(i) - \bar{O}(i)]^2}{\partial r(w,p)} = 2\sum_{i\in I}\left\{[O(i) - \bar{O}(i)] \times \frac{\partial O(i)}{\partial r(w,p)} \times \mu^i_{w,p}\right\} \quad (6)$$

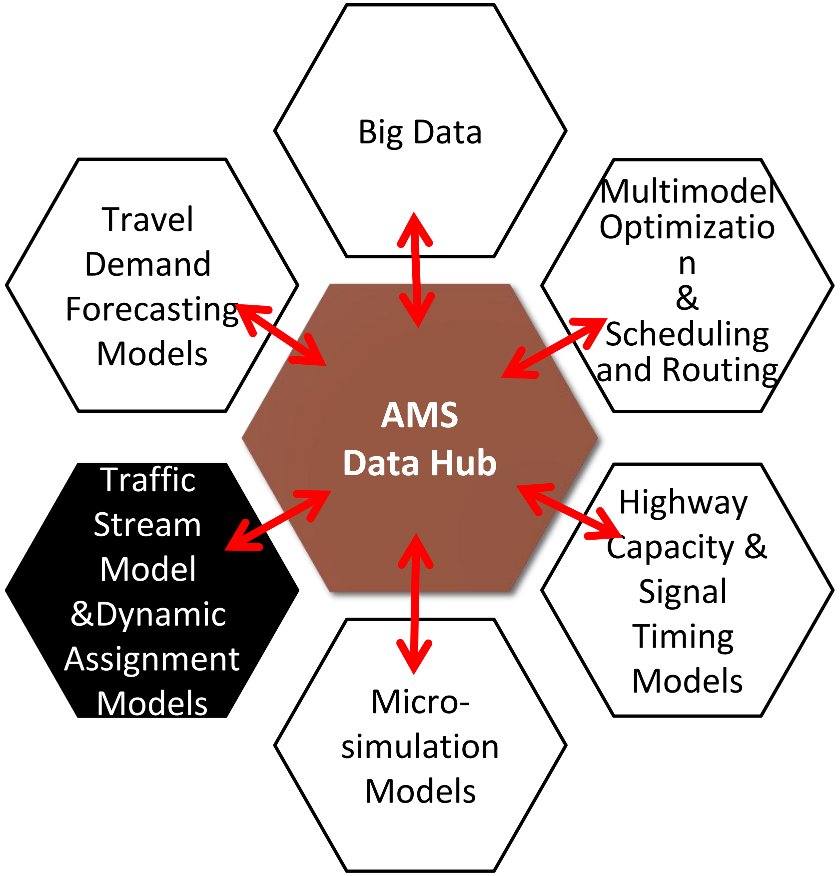
$$\nabla h^d(r) = \frac{\partial \sum_{j\in J} [D(j) - \bar{D}(j)]^2}{\partial r(w,p)} = 2\sum_{j\in J}\left\{[D(j) - \bar{D}(j)] \times \frac{\partial D(j)}{\partial r(w,p)} \times \lambda^j_{w,p}\right\} \quad (7)$$

2.4 City, Land Use and Urban Layout

https://github.com/xzhou99/Urban-Layout-QAP



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/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

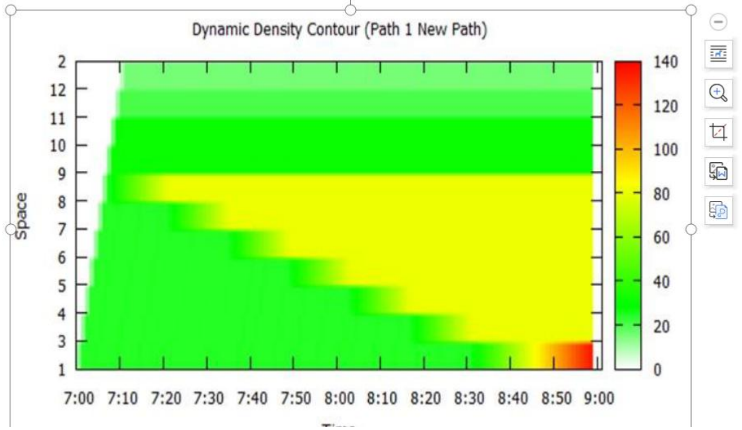
3.1 Traffic flow modeling

https://github.com/xzhou99/stalite-dtalite_software_release/tree/gh-pages/learning_document/3_dynamic_user_equilibrum_and_traffic_bottleneck

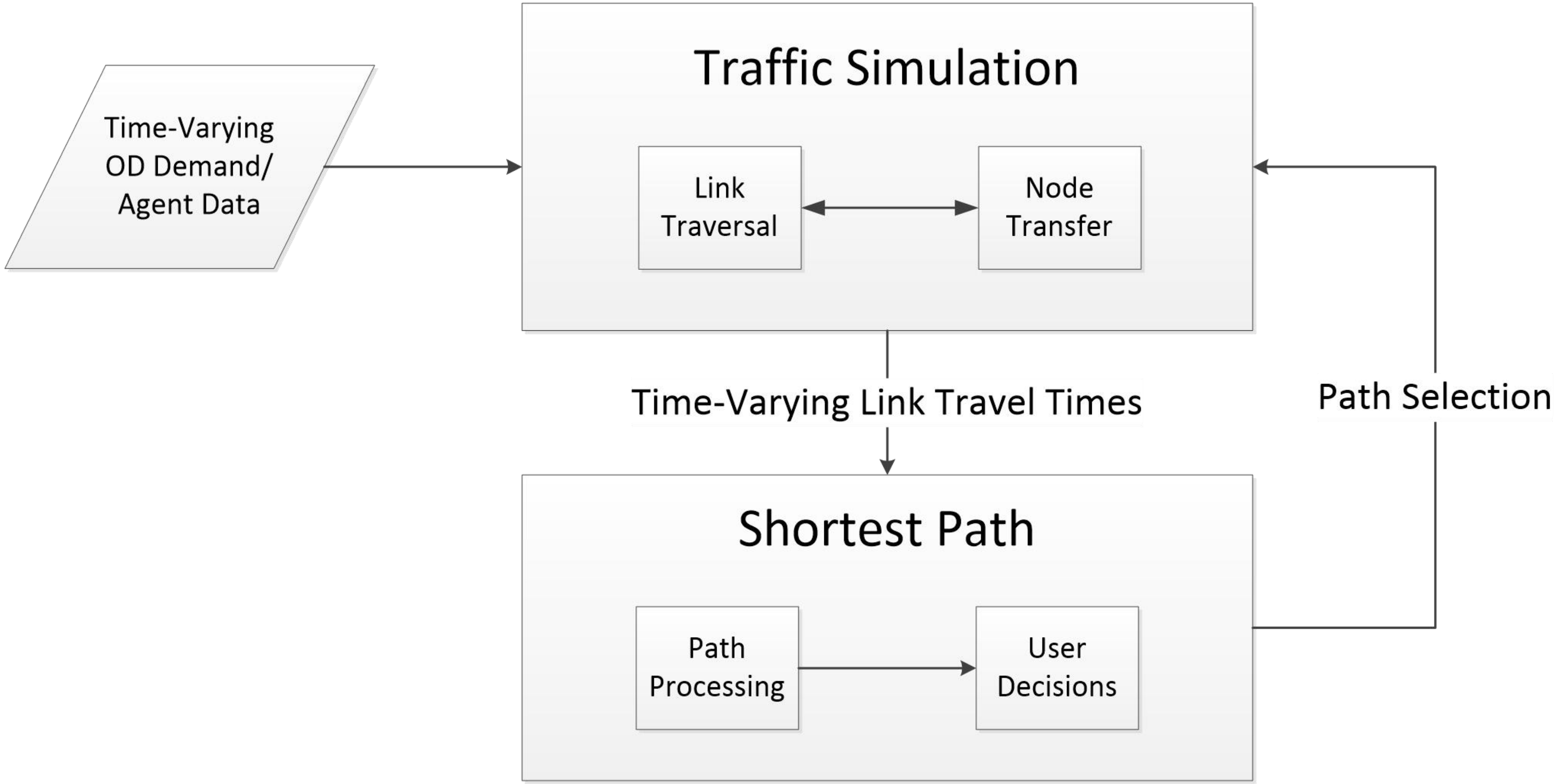
1. BPR travel time functions
2. Point queue (relaxed storage constraints)
3. Spatial queue (similar to DYNASMART-P/Dynus-T model)
4. Newell's model (similar to Cell Transmission Model)

    Shockwave propagation

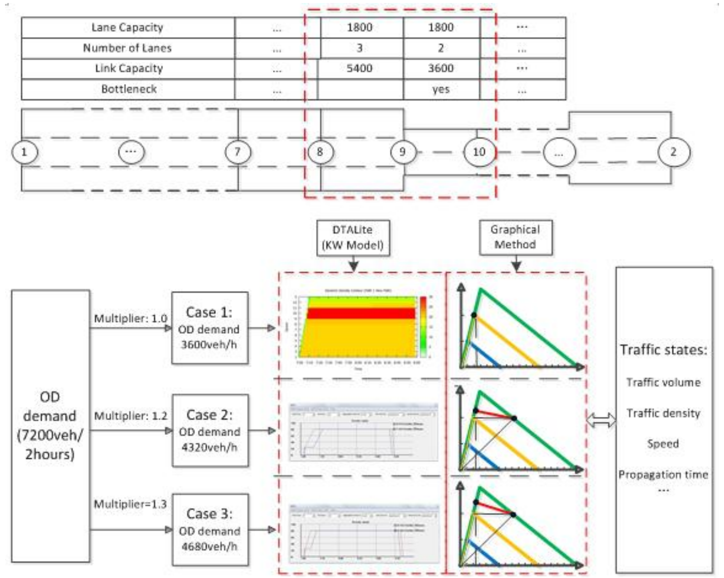
Traffic Congestion Propagation: Understanding Theoretical Basics of Dynamic Traffic Network Assignment and Simulation

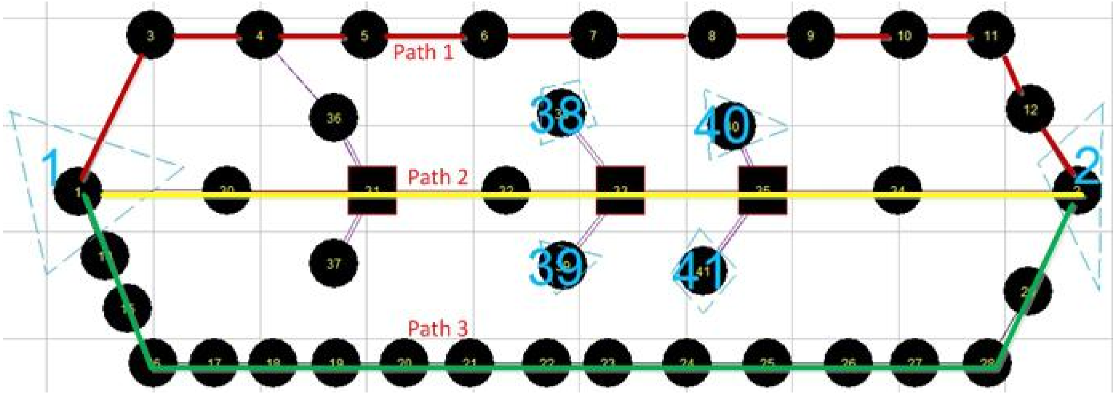


3.2 Understand Traffic Simulation



3.3 Understand dynamic traffic equilibrium

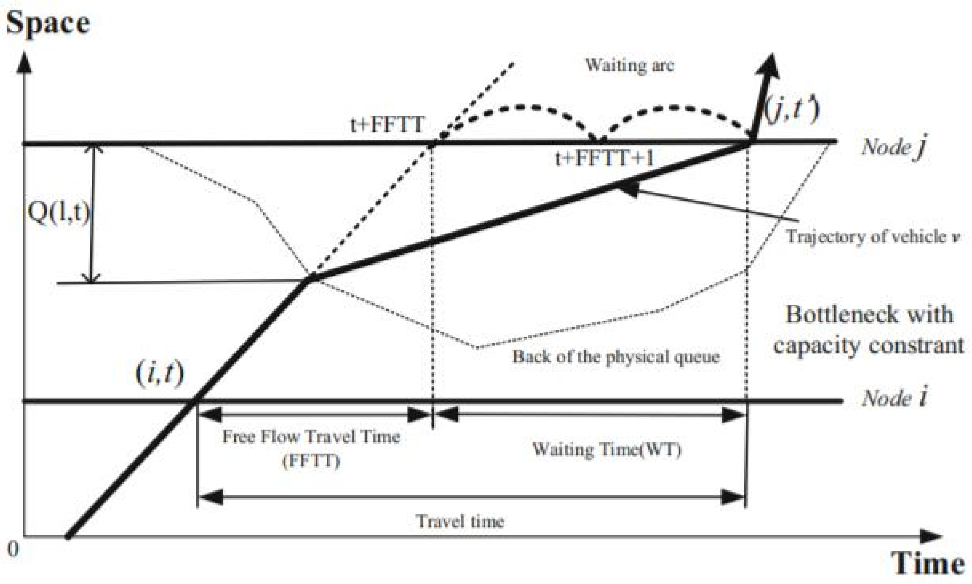


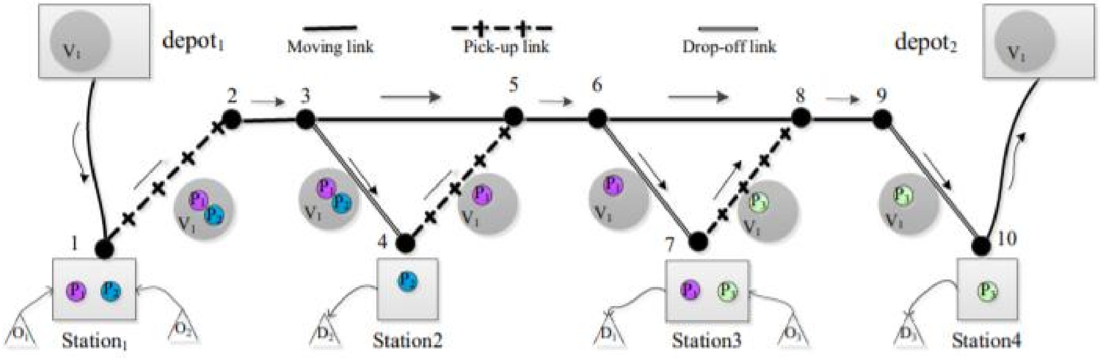


3.4 Public Transportation Mobility Simulation for Bridging Multi-agent Simulation and Optimization

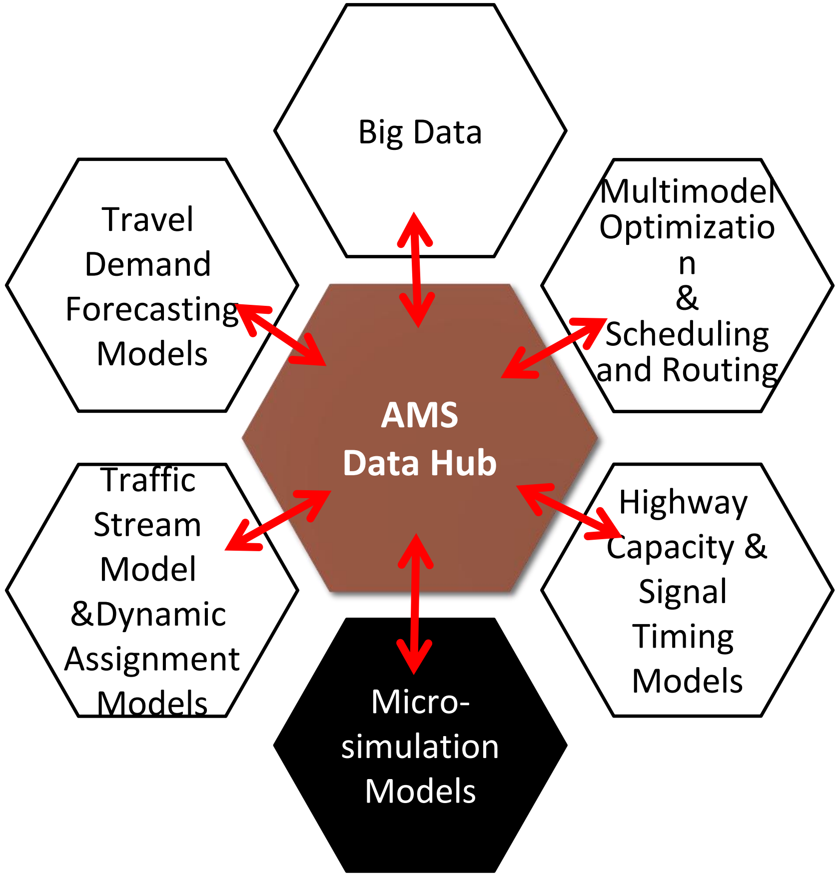
https://github.com/xzhou99/DTALite-S

https://link.springer.com/article/10.1007/s40864-018-0100-x





Connection 4: Microsimulation Models



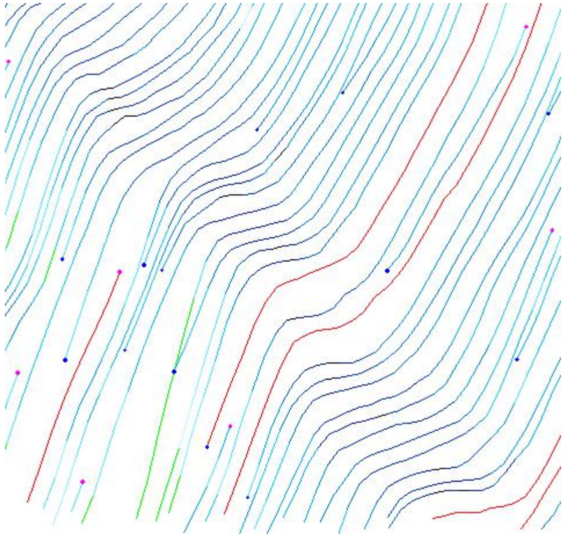
•https://github.com/xzhou99/NeXTA_4_NGSIM_Trajectory_Visualization

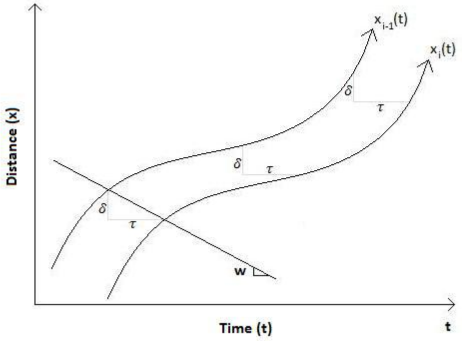
- Multi-resolution network convertoer: Ocean https://github.com/jiawei92/Ocean

- 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

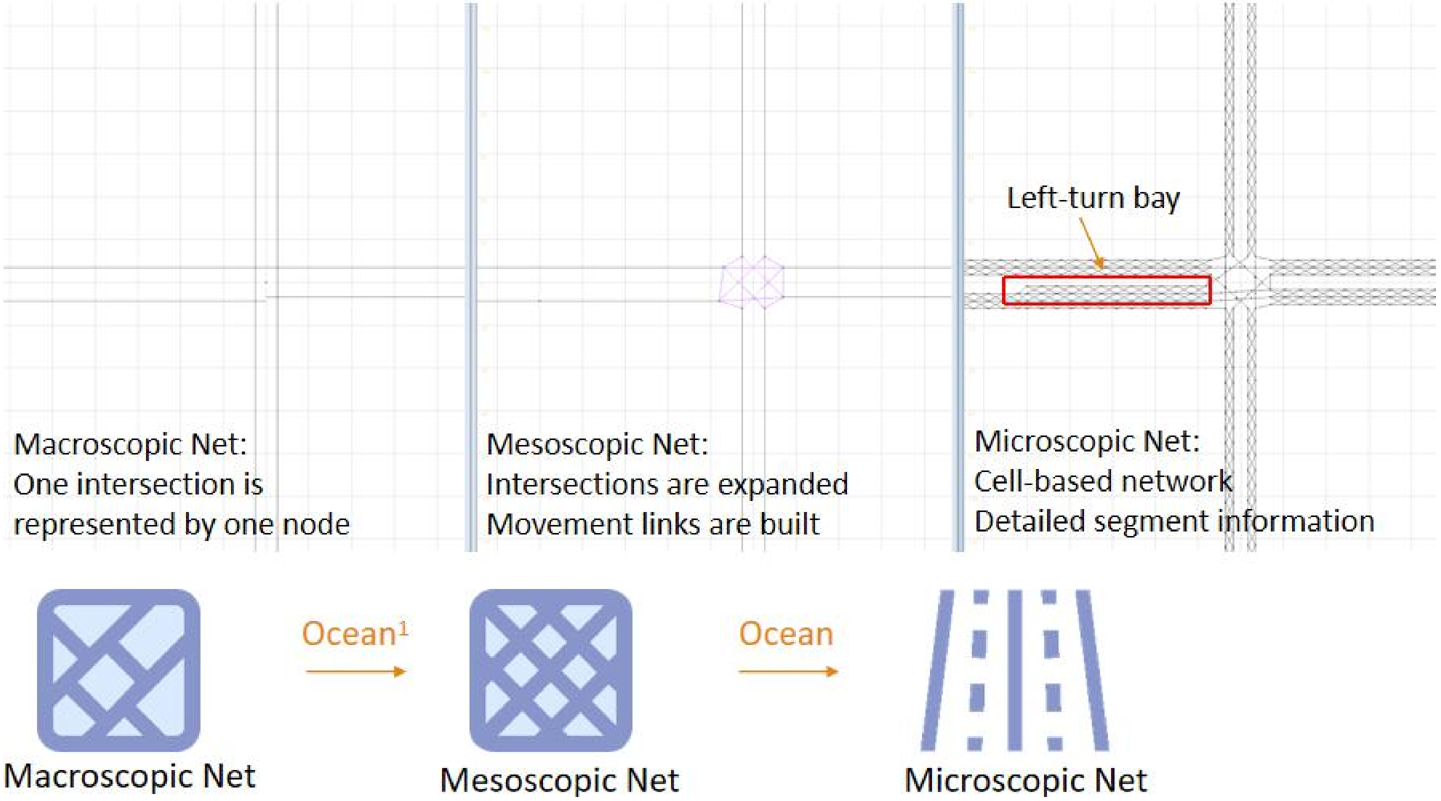




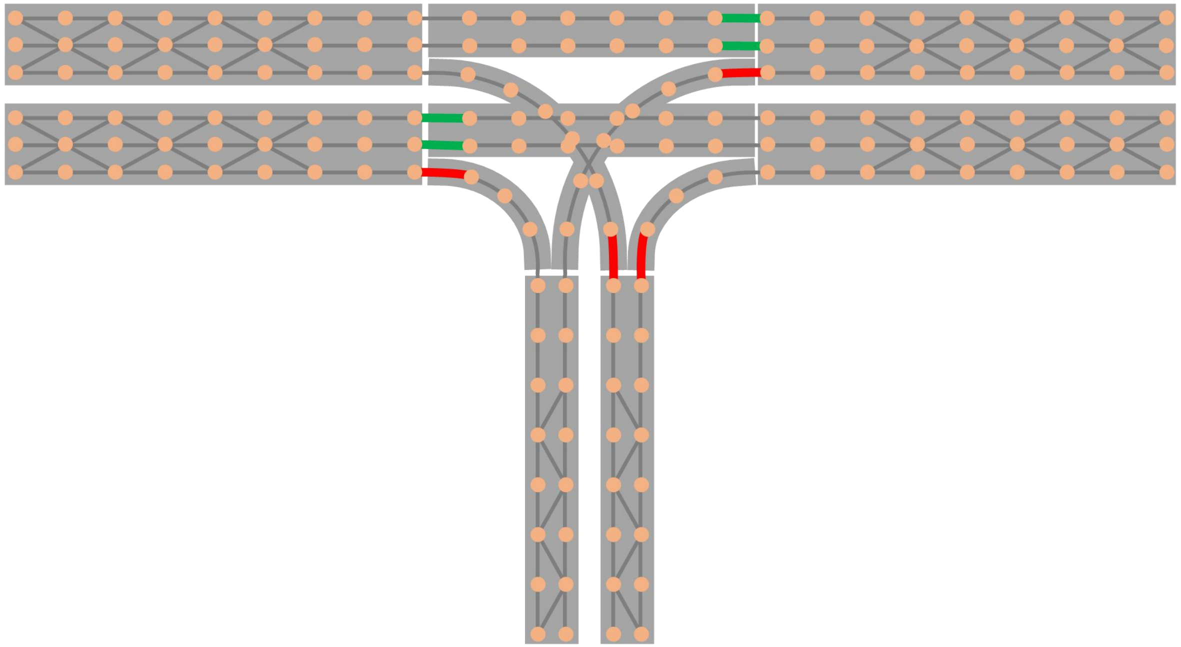
Reaction distance/spacing δ
Reaction time lag τ
$W = \delta / \tau$

4.2 Building multi-resolution networks

https://github.com/jiawei92/Ocean



4.3 Modeling Signal Control Logic



4.4 Multi-resolution Traffic Modeling

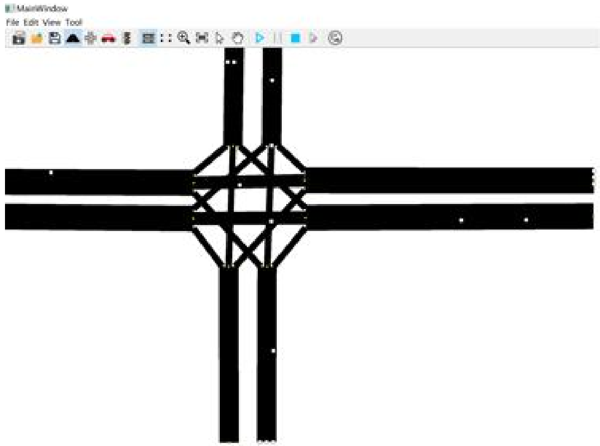
Network Preparation → Network Downloader, Ocean

Demand Preparation → STALite (ODME)

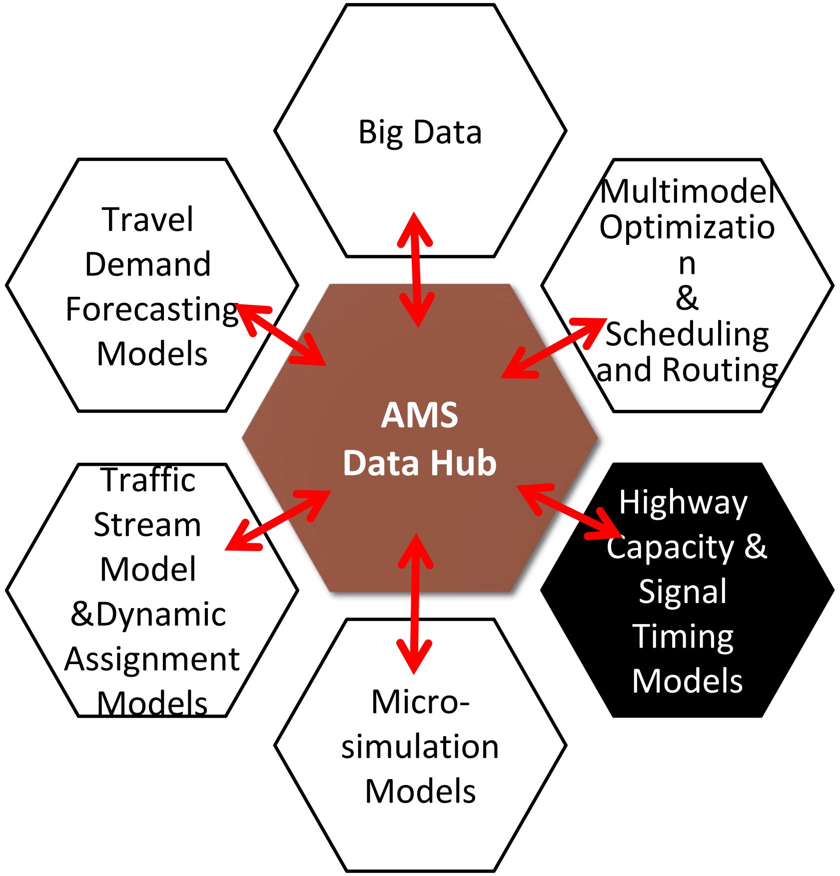
Calibration → Activity Graph

Simulation → CAVLite

Visualization & Output → CAVLite GUI



Connection 5: Capacity Analysis Tools

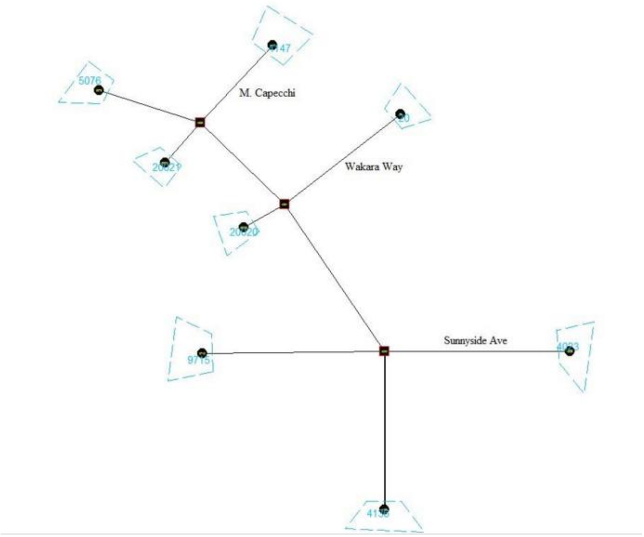


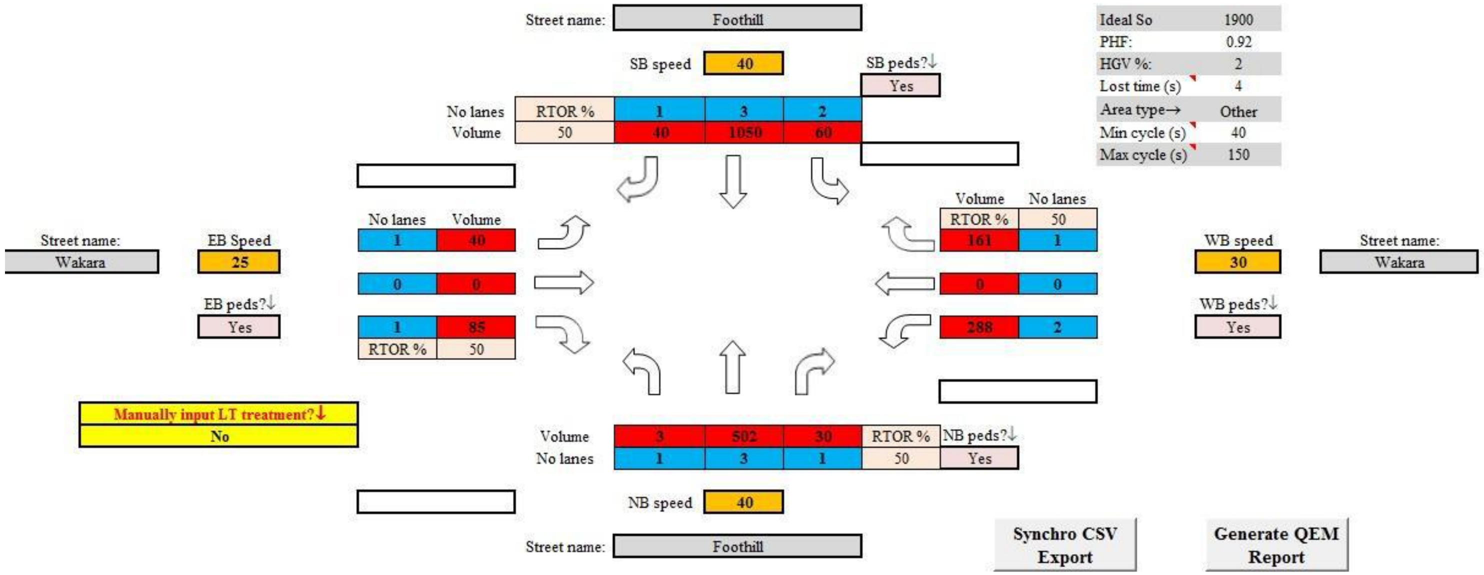
- HCM-based Quick Estimation Method (QEM)
  - Given turning volume,
  - Generate signal phasing and timing data
  - https://github.com/xzhou99/Sigma-X
  - https://github.com/xzhou99/SignalAPI

- 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

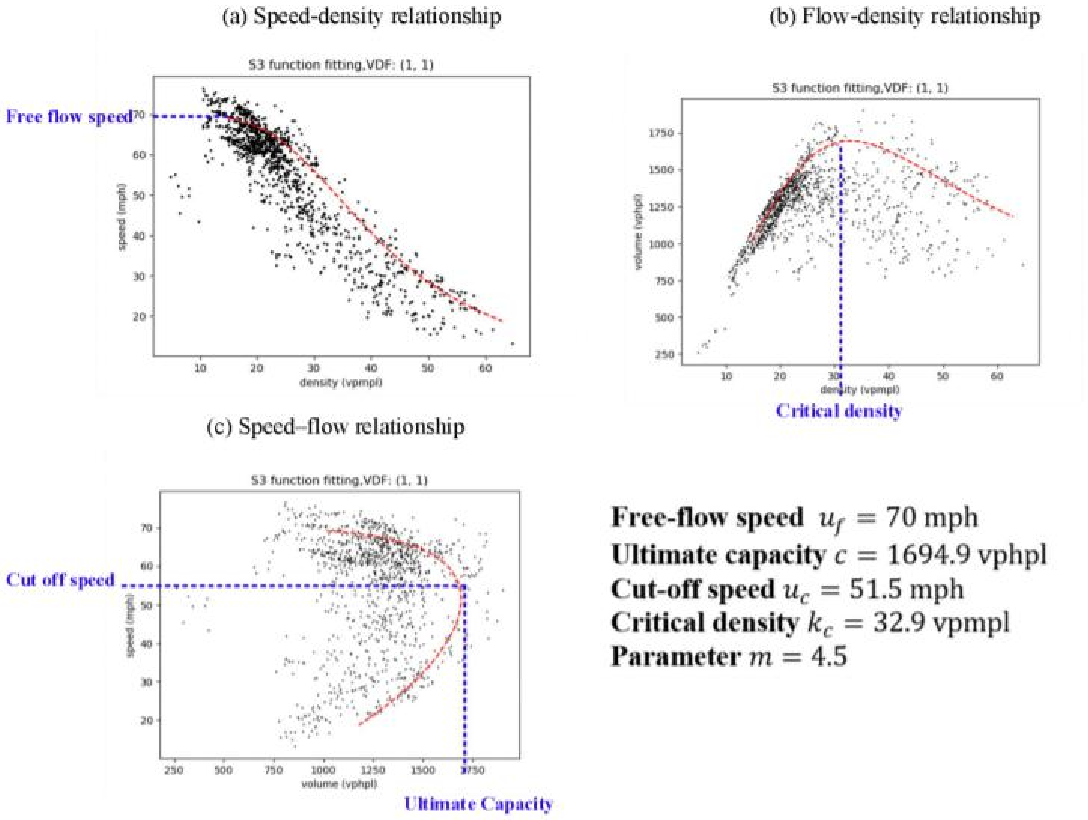
- https://github.com/xzhou99/Sigma-X
- https://github.com/xzhou99/SignalAPI





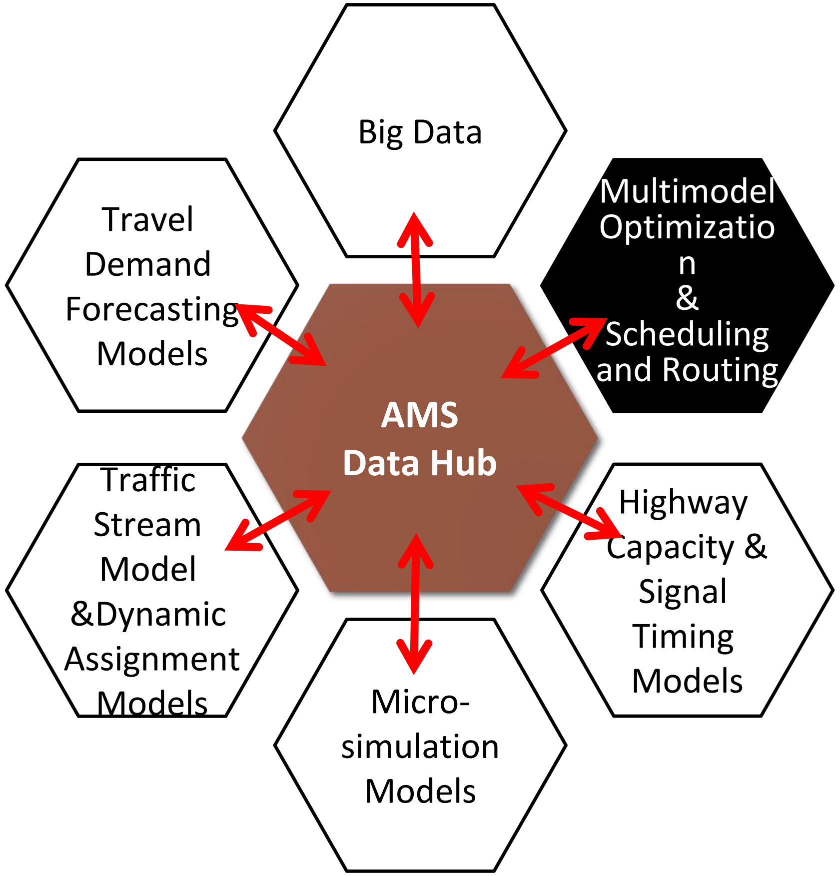
5.2. Freeway capacity analysis

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 $m = 4.5$

Connection 6: Optimization, Routing and Scheduling

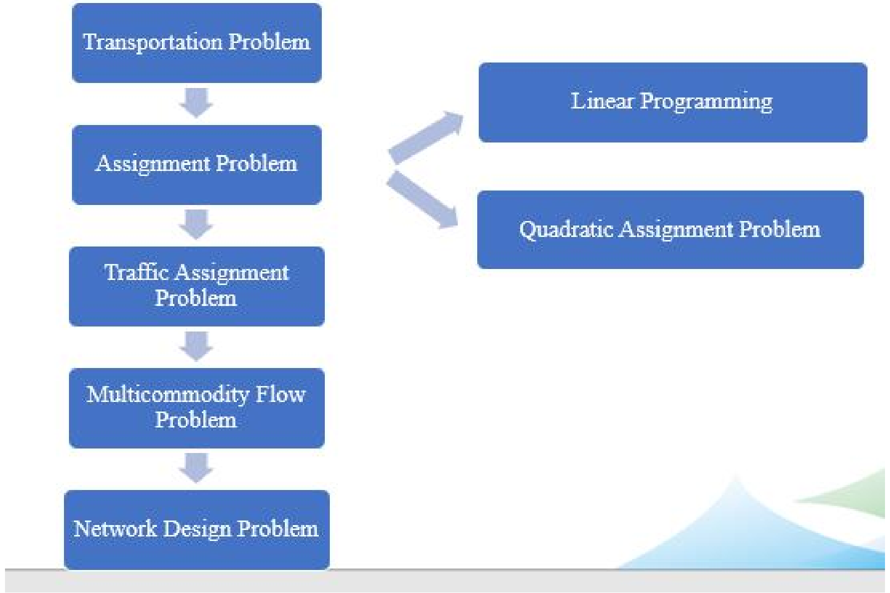


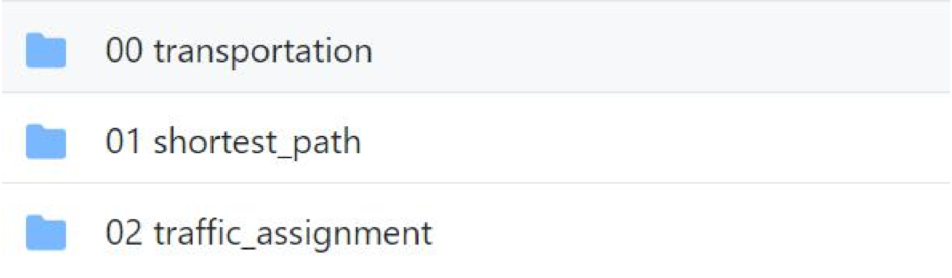
- Optimization
  - Transportation problem
  - Shortest path problem
  - Space time network design
  - Transit network design
  - Yard management
  - Vehicle routing problem
  - Train scheduling
  - Customized bus service design
- https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

- 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

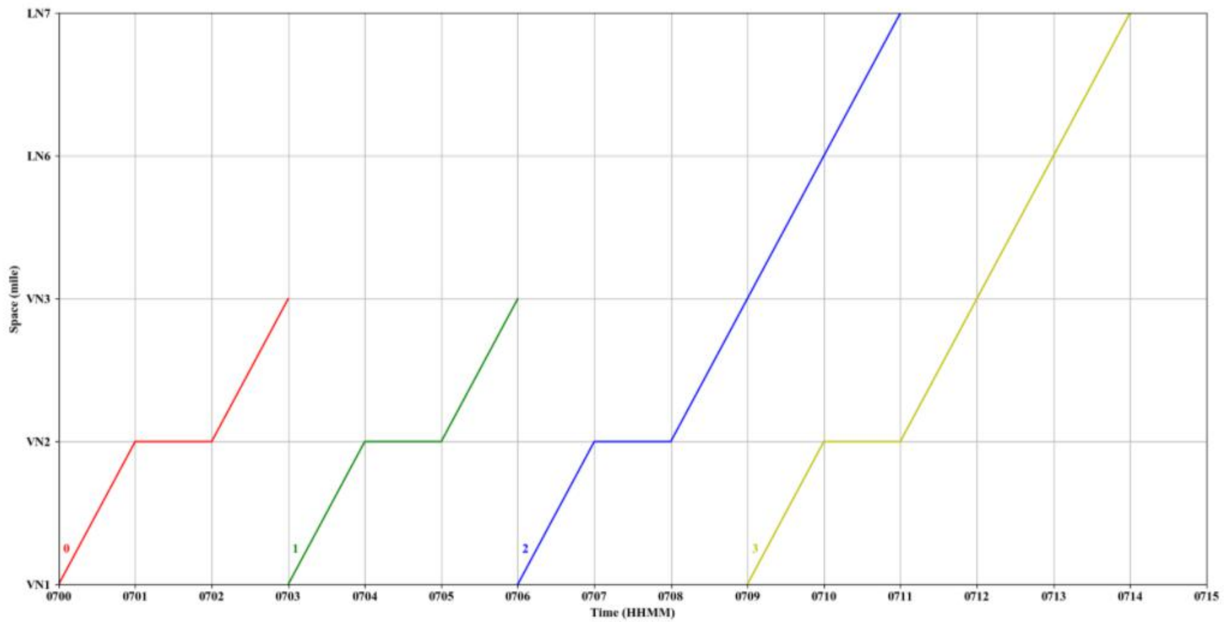
- https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network





6.2 From space-time diagram to space-time networks

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



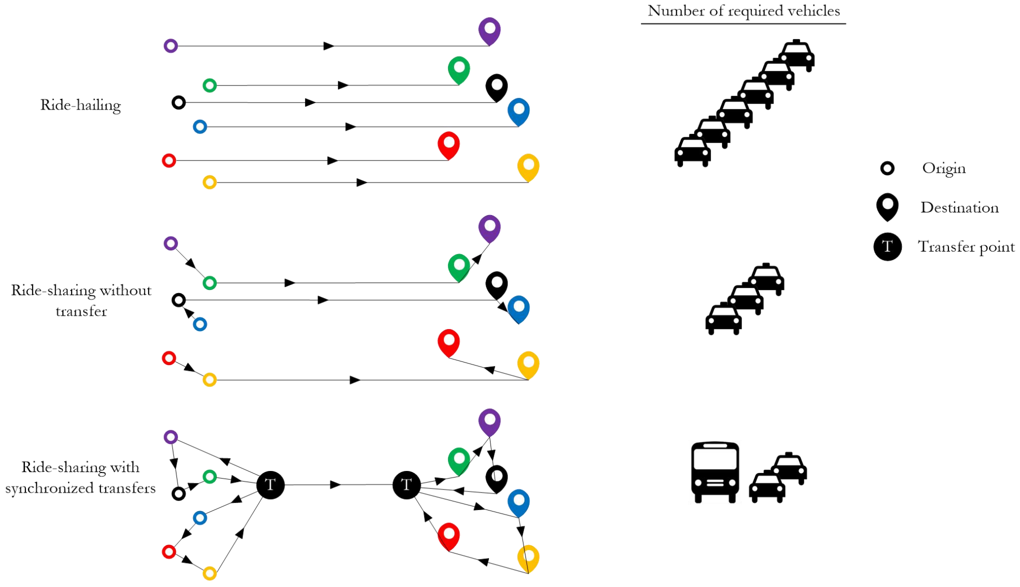
03 space_time_network_design_for_accessibility

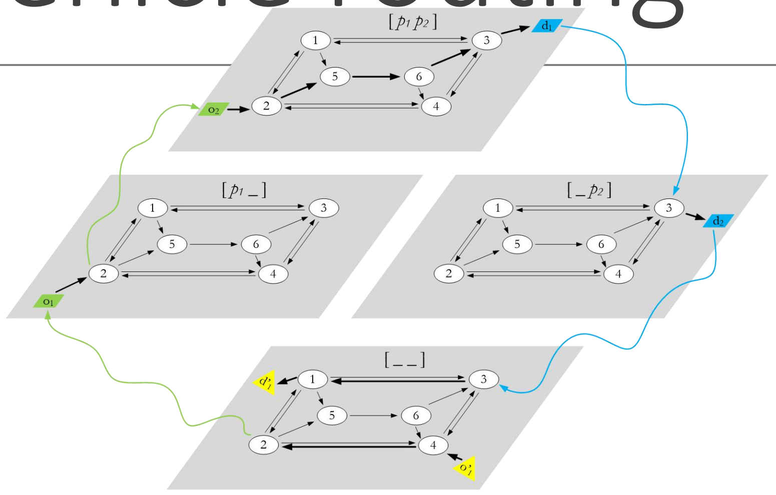
04 transit_network_design_and_branch_and_bound

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

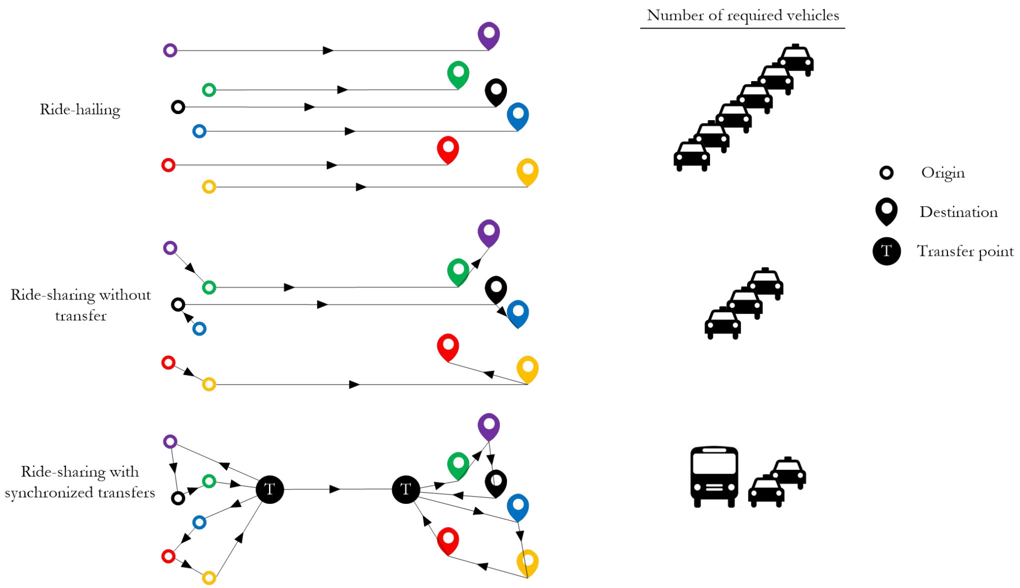




https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network

- 05 Vehicle Routing Problem with pickup and delivery...
- 06 space_time_speed_network_for_train_schedudling

6.4 Use space-time network in multimodal system scheduling



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

- 07 Customized_bus_service_design
- 08 train scheduling for minimizing passenger waiting...
- 09 multi-layer rail yard_management

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:

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

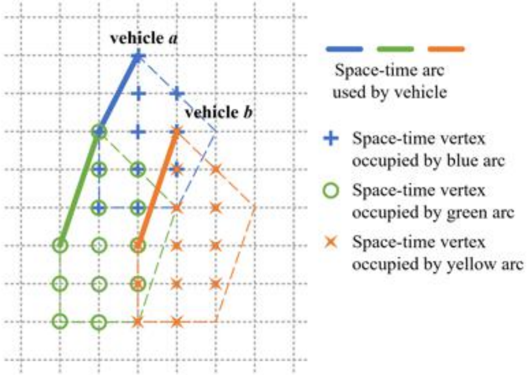
Indicator of vertex visited by vehicles:

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

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

- 11 Traffic OD demand estimation
- 12 household activity scheduling

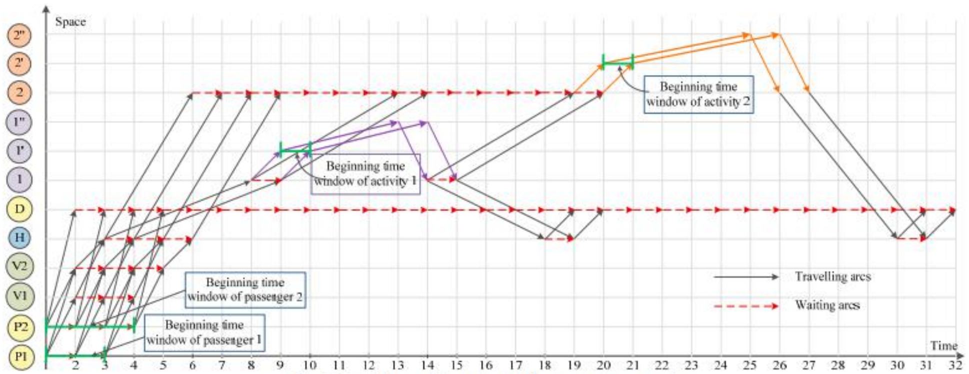


Fig. 3 Feasible searching region in the space-time network

Objective function

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

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

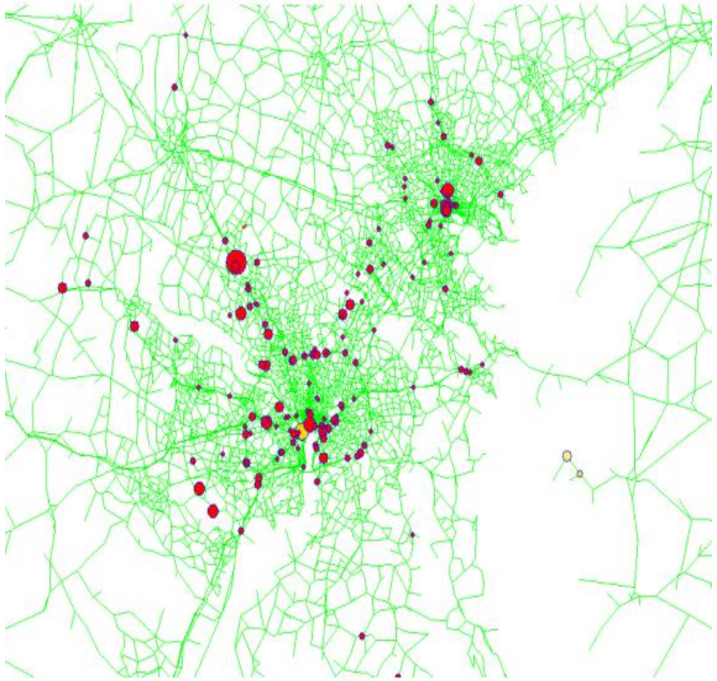
NEXTA

Traffic state estimation and prediction, Train routing and scheduling

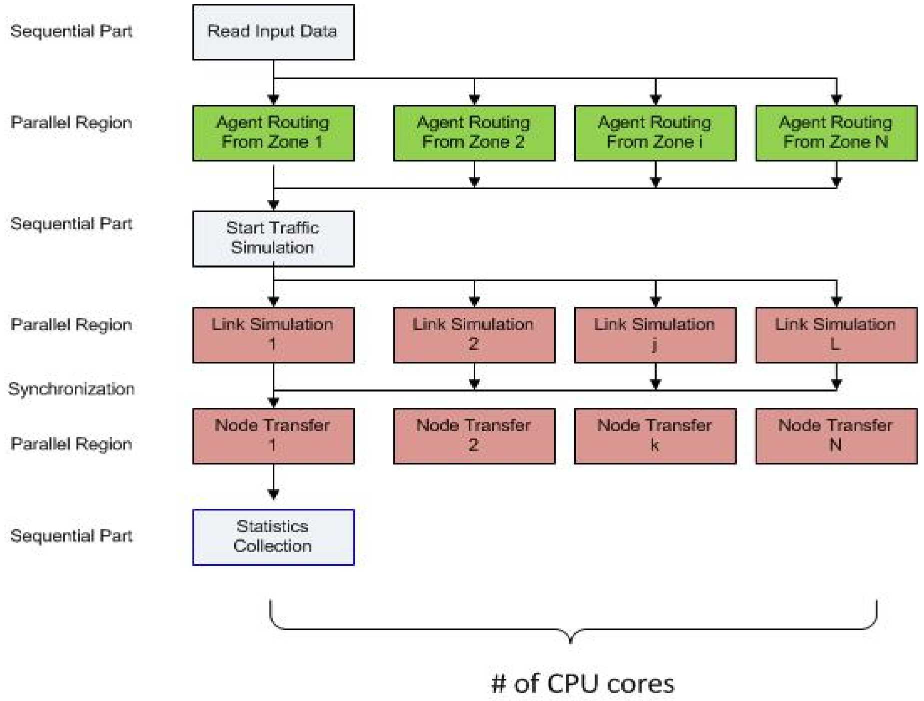
VRPLite

5. Large scale applications

Computational Challenges

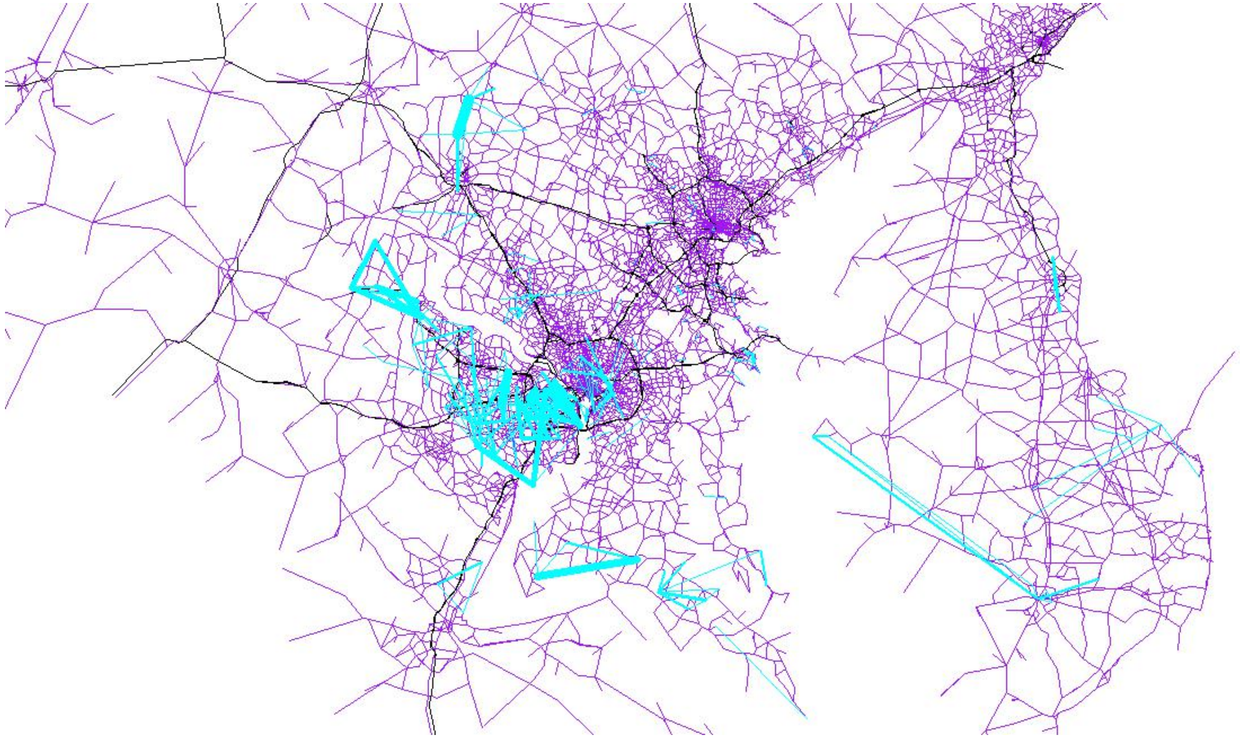


Shared memory-based parallel computing for agent-based path finding and mesoscopic traffic simulation (based on OpenMP)



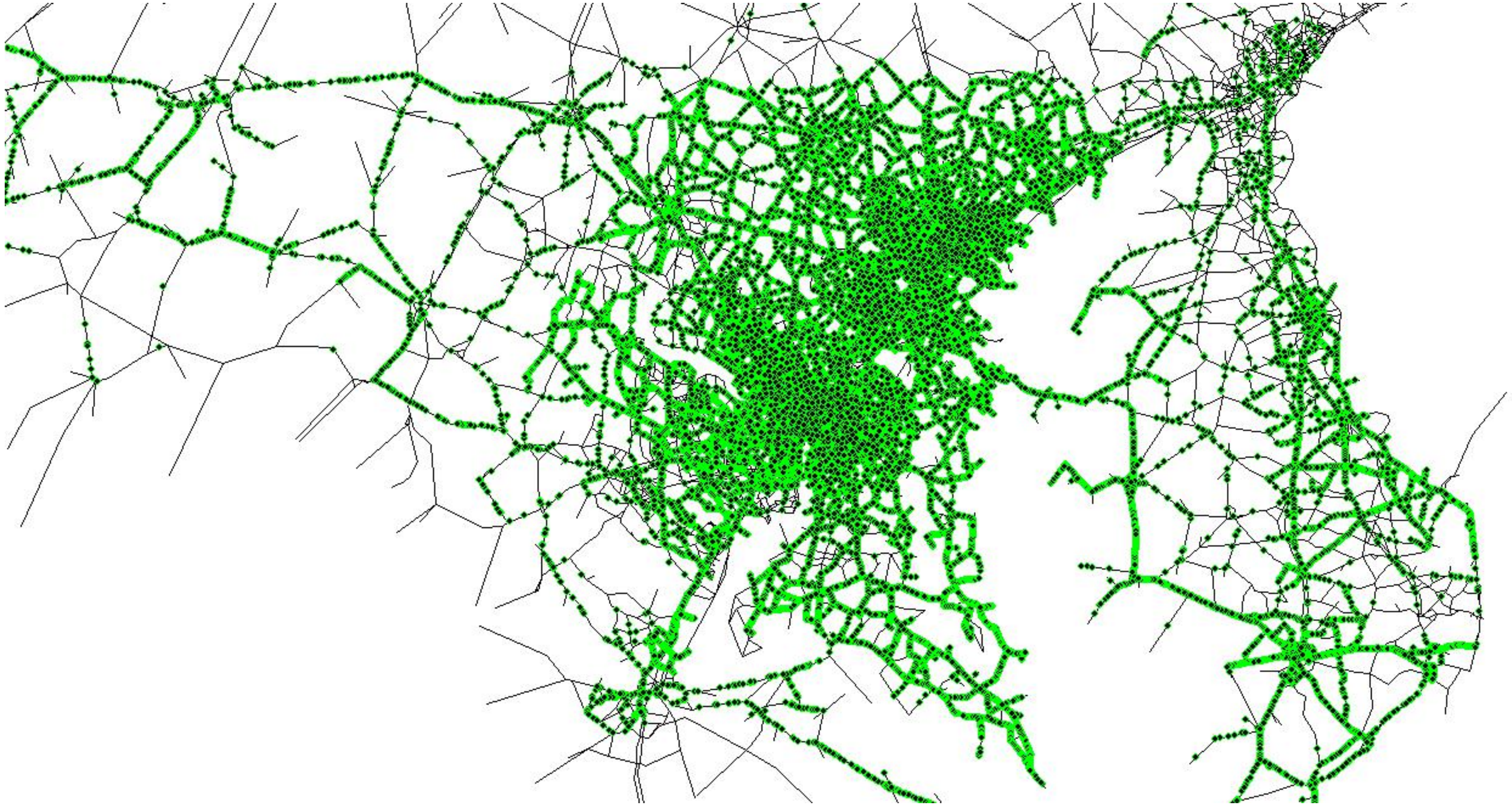
Maryland State-wide model:
20 K nodes, 47K links, 3,000 zones, 18 M agents
CPU time: 30 min per UE iteration on a 20-core workstation with 194 GB RAM

Origin-Destination Demand Spatial Distribution Pattern

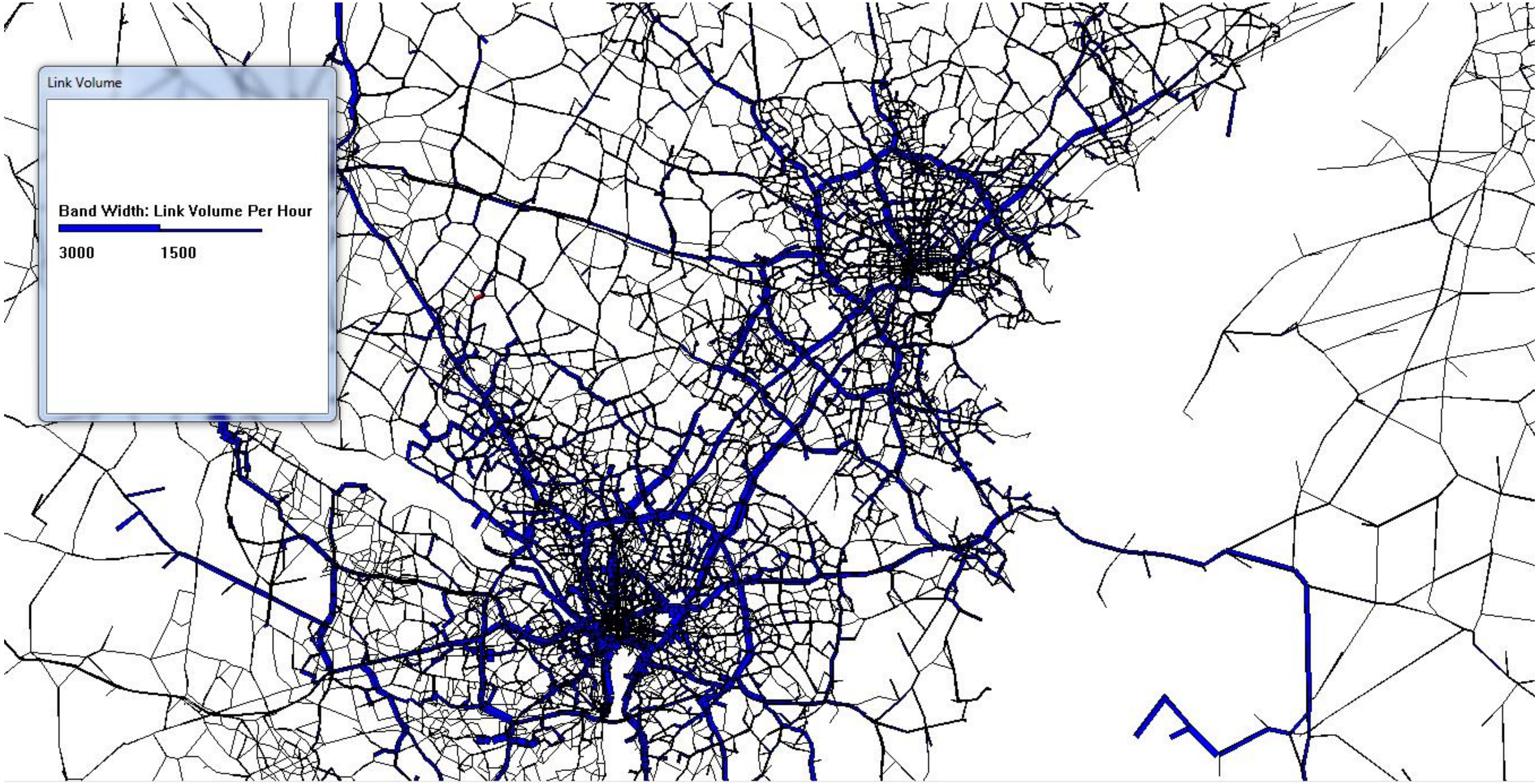


Collaboration with University of Maryland and Maryland State Highway Administration
Supported by TRB SHRP II Program

Vehicle Animation at Network Level

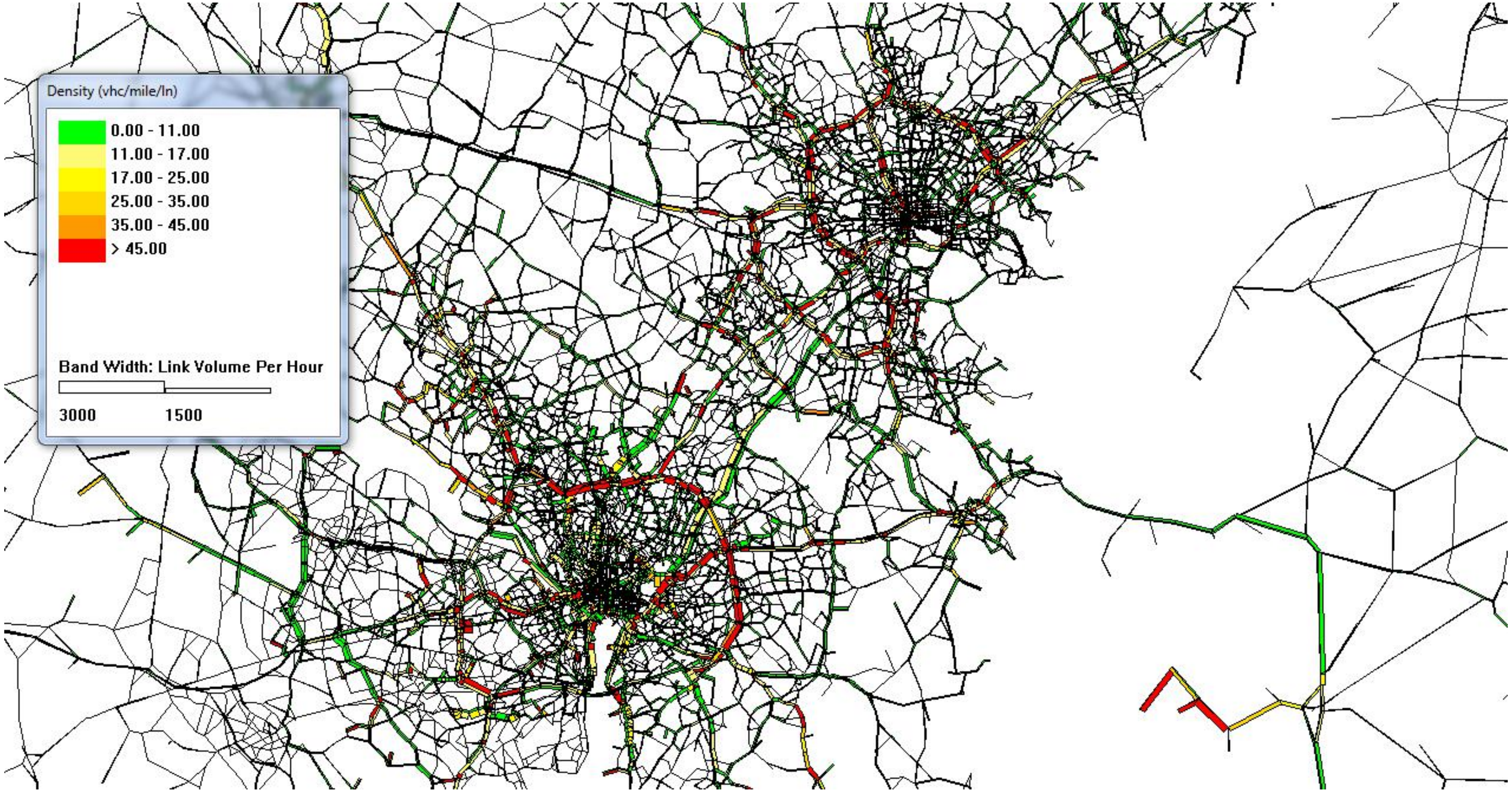


Volume at Network Level

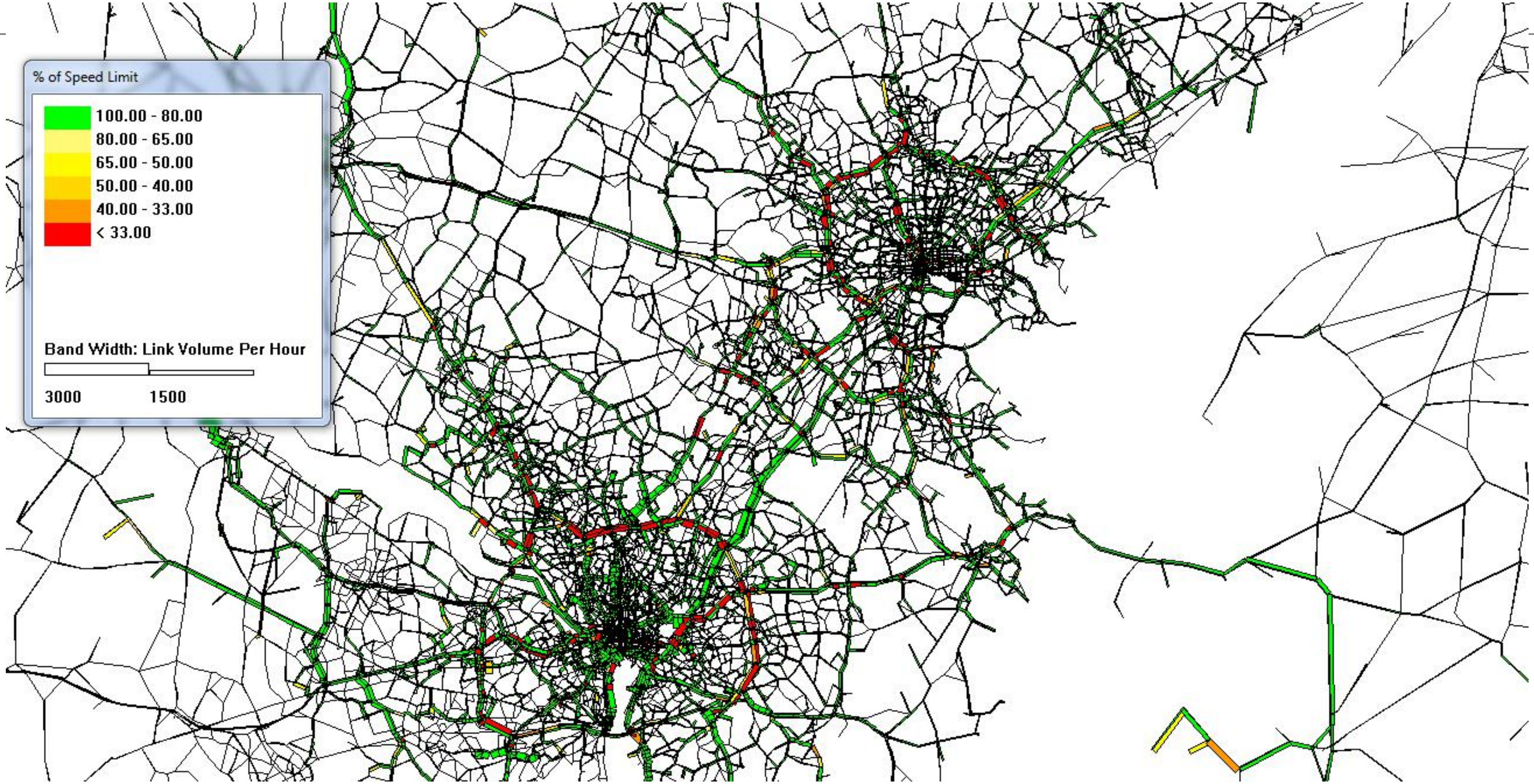


Band width of a link is proportional to link volume

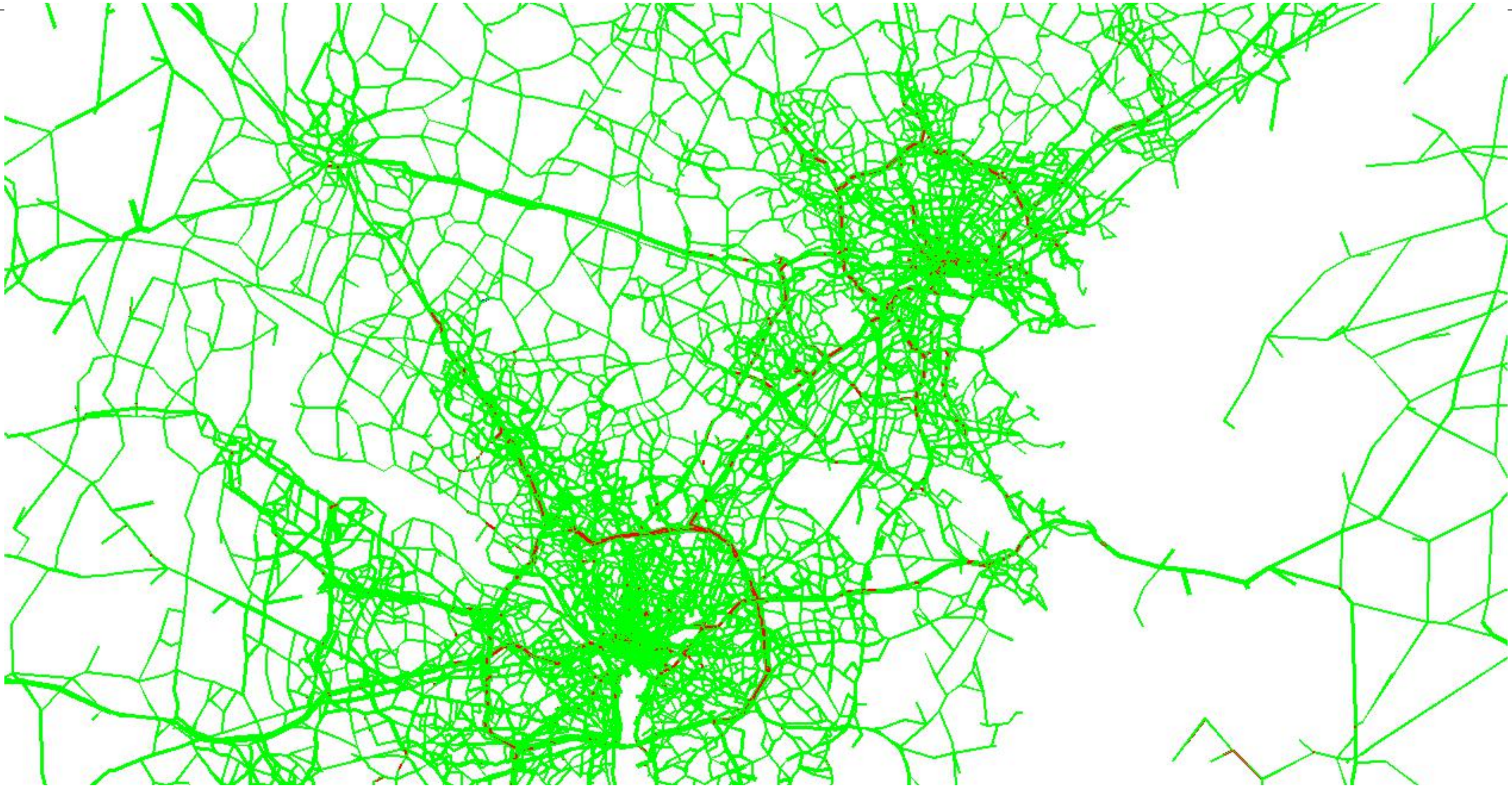
Density at Network Level



Speed at Network Level



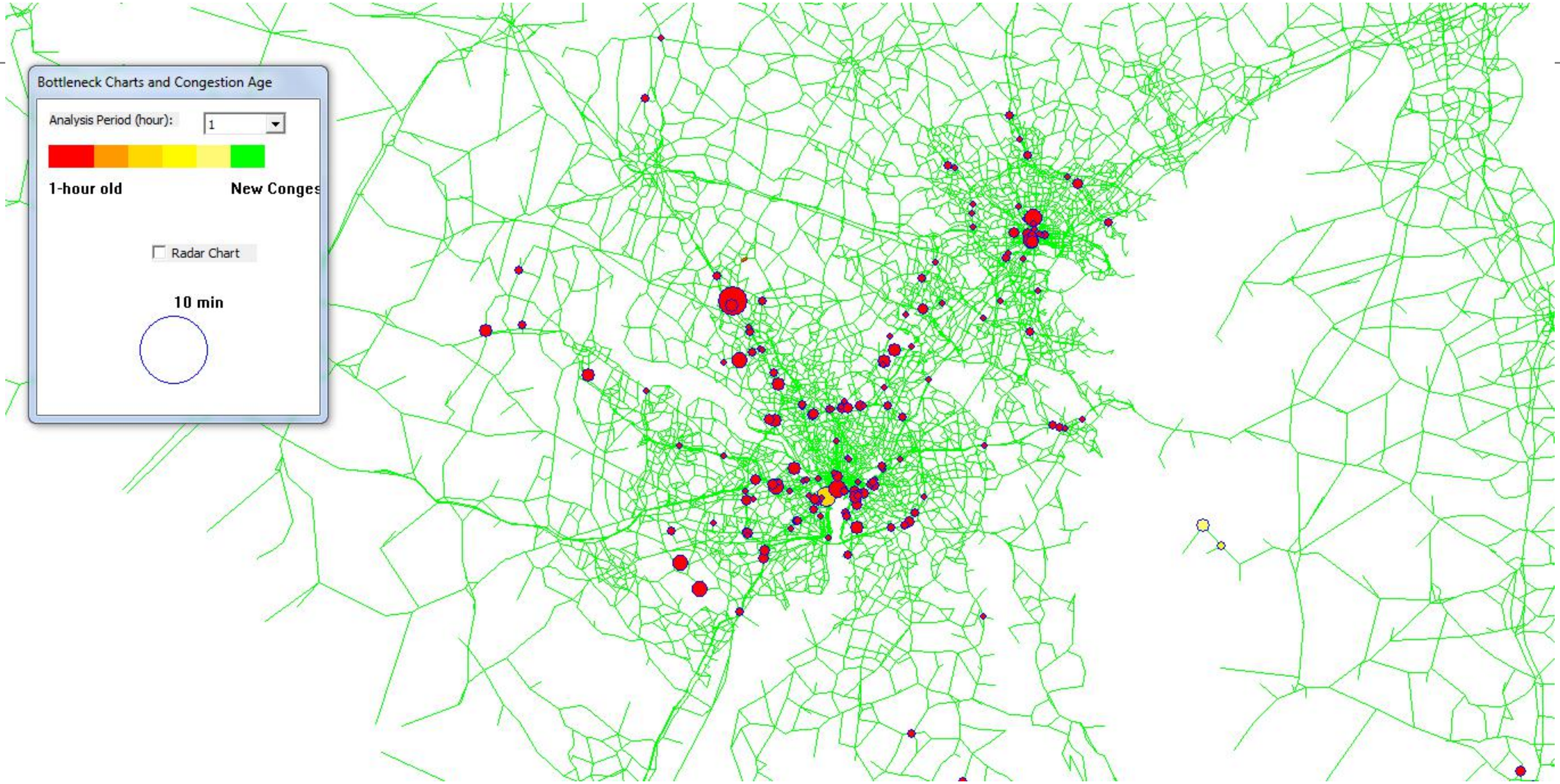
Queue at Network Level



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