# Open-Source Dynamic Traffic Assignment Package

# DTALite/NEXTA

**Xuesong (Simon) Zhou**

xzhou99@gmail.com

xzhou74@asu.edu

# Workshop Purpose and Objectives

**Purpose**

- Educate workshop attendees on Dynamic Traffic Assignment modeling through hands-on exercises using the open-source DTALite software tool

**Learning Objectives**

- Understand basic modeling approaches in DTALite
- Import data to code and modify a subarea network
- Evaluate simulation results using the visualization and reporting features in NeXTA

# Outline

Module 1: Introduction to DTALite/NeXTA

Module 2: Introduction to DTA modelling principles

Module 3: Visualization Features in NeXTA

Module 4: Introduction to Workshop Exercises

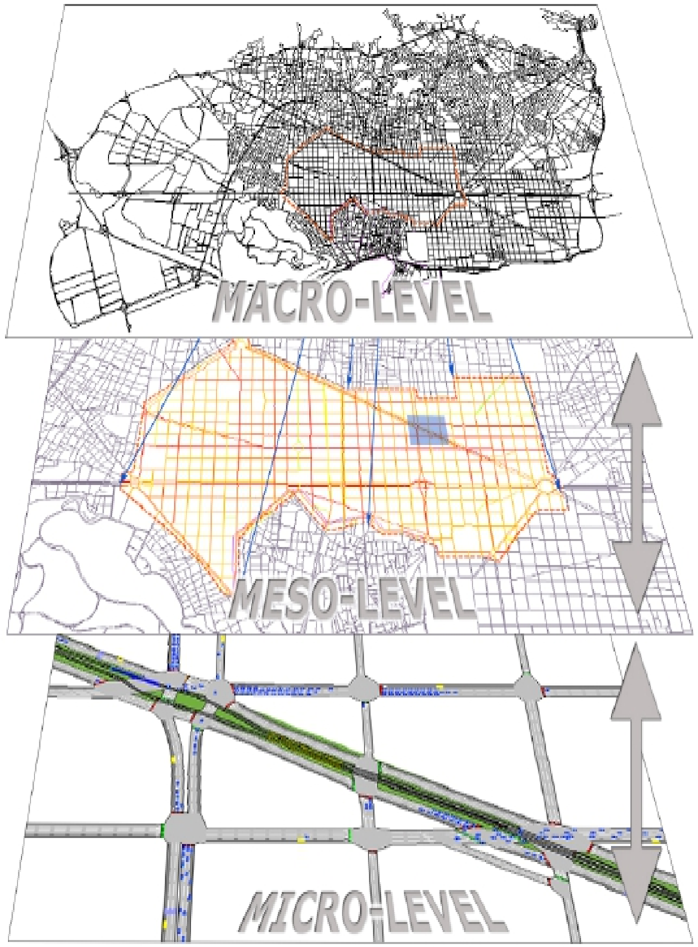
# Module 1
# Introduction to DTALite/NeXTA

1.1 Project Objective and Motivations

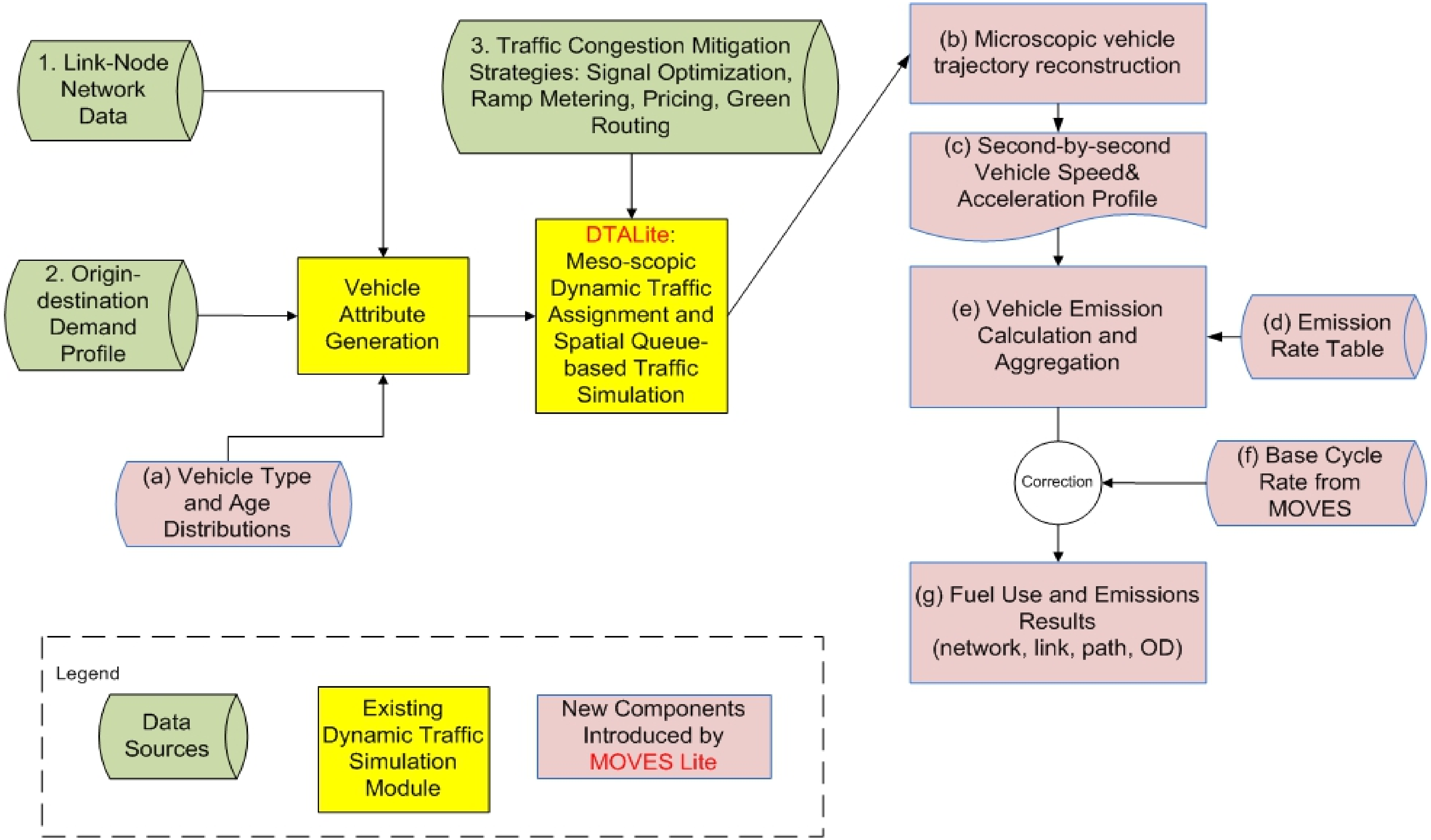
1.2 What NeXTA/DTALite Can Do

1.3 Open-source Free Software Package

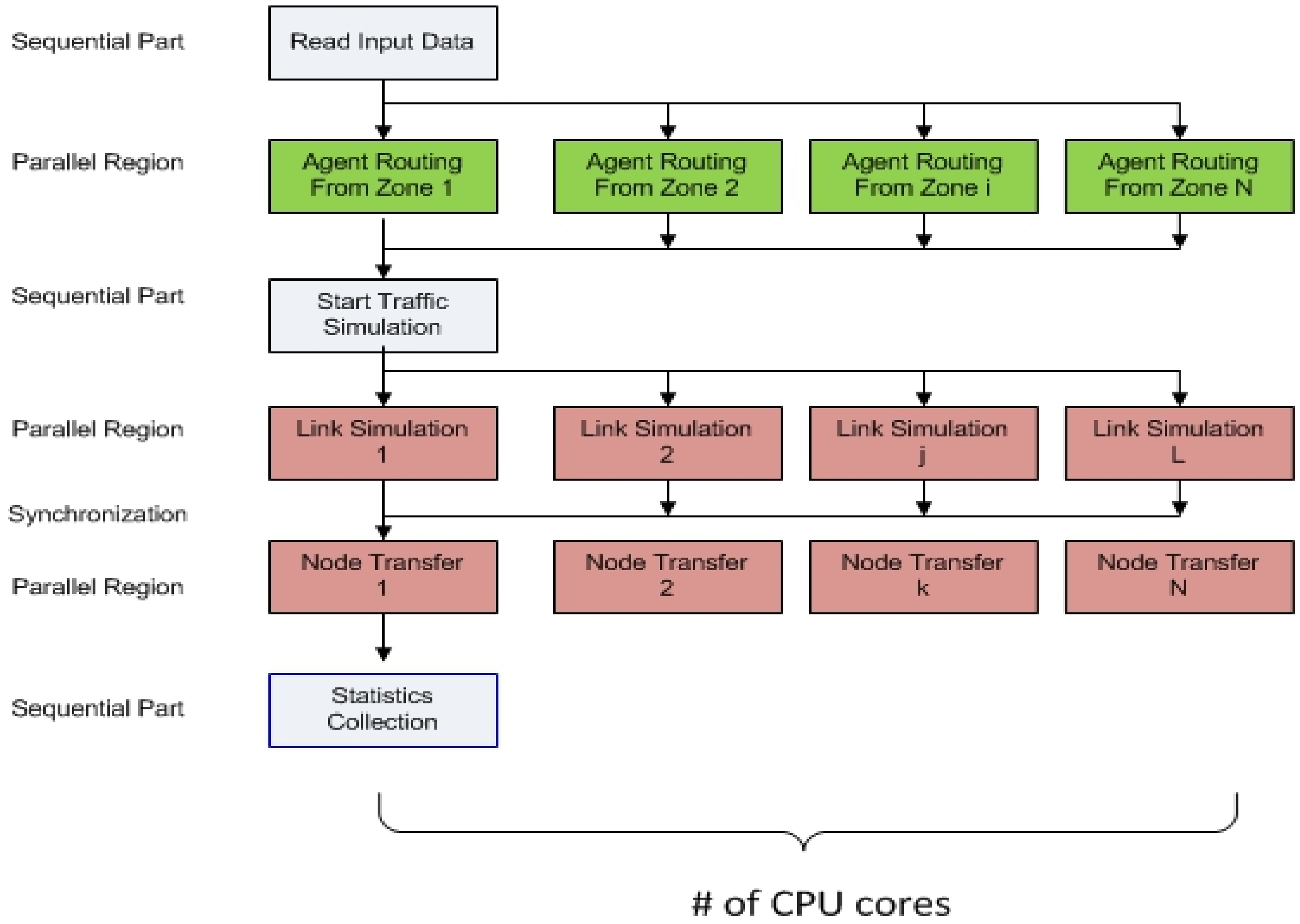
# Mesoscopic Traffic Simulation



# System Architecture and Data Flow



# Computational Efficiency



# 1.0 Traffic Flow Theoretical Foundation: LWR Equations

- Hyperbolic system of conservation laws

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = g(x,t)$$

  where q, k are flow and density, respectively, and g(·) is the net vehicle generation rate.

- Traffic flow model
  - Speed (or flow) of traffic is a function of density only
  - E.g. Greenshields model: $V = V_{free}\,(1 - k/K_{jam})$

# Numerical Calculation Scheme Based on Link Flow/Density (Used in CTM)

- Notation
  - $\Delta t$ = length of simulation interval (e.g. 1 sec, 6 sec)
  - $\Delta x$ = cell length (e.g. 0.5 mile)
  - $k_{i,t}$ , $v_{i,t}$ =the prevailing density and mean speed in cell i during the $t^{th}$ time step
  - $q_{i,t}$ = transfer flow rate from section i to cell i+1 during the $t^{th}$ time interval $[t, t+\Delta t]$.
- Computation Procedure
  - Step 1: Calculate prevailing speed $v_{i,t}$ according to a traffic flow model
  - Step 2: Calculate flow ready to move from cell i-1 to section i: $v_{i,t} \times k_{i,t}$
  - Step 3: Calculate transfer flow from section i to cell i+1

  $$q_{i,t} = \text{Min}\ \{ v_{i,t} \times k_{i,t}\ ,\ q_{max\ i+1,t} \}$$

  - Step 4: Update prevailing density at cell i

  $$k_{i,t+1} = k_{i,t} + (q_{i-1,t} - q_{i,t}) \times \Delta t / \Delta x$$

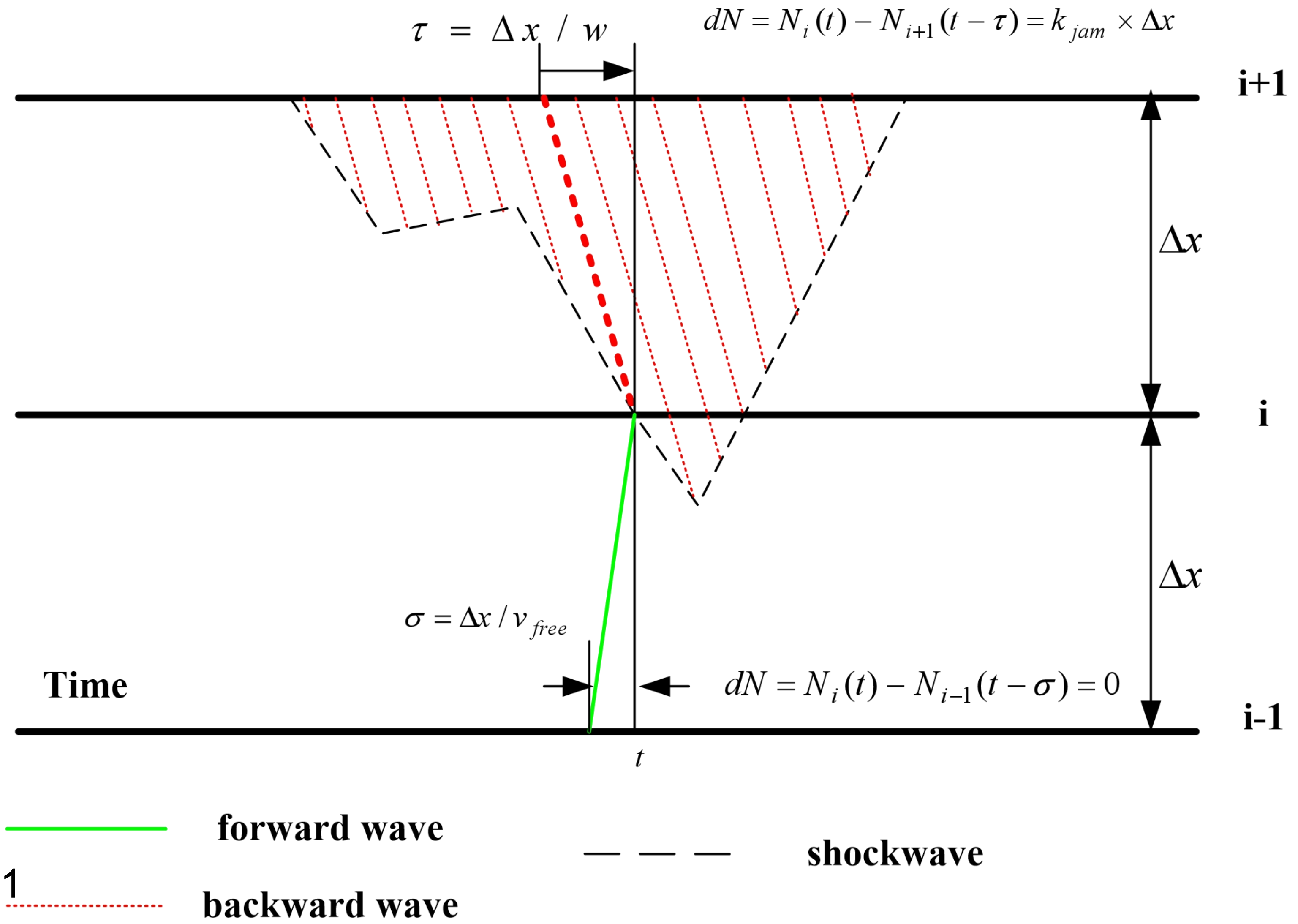
- Issue: transfer flow $q_{i,t}$ is an explicit function of the occupancy at the current cell, other than the downstream capacity
  - Transferred flow exceeds the spatial storage capacity at the next cell

# Newell's N-Curve Approach (Used in DTALite)

- $N_i$ (t): cumulative flow counts at location i at time t
  - $q=[N_i(t+1)-N_i(t)]/\Delta t$
  - $K=[N_{i+1}(t)-N_i(t)]/\Delta x$
- Focus on N-curve at transfer points
  - $N=\mathrm{Min}\ \{N_{upstream}, N_{downstream}\}$
- Focus on change of N along a characteristic line (wave) at sections
  - $dN = (\partial N/\partial x)dx + (\partial D/\partial t)dt = -\ kdx + qdt =$

    $(-k+ q \times \text{wave})dx$
  - if wave=$v_{free}$, $dN=( -k + k)\ dx=0$
  - if wave=w, $dN=-(k+qw)dx= -k_{jam}dx$

# Illustration of N-Curve Computation For Tracking Queue Spillback



# Summary of Comparison

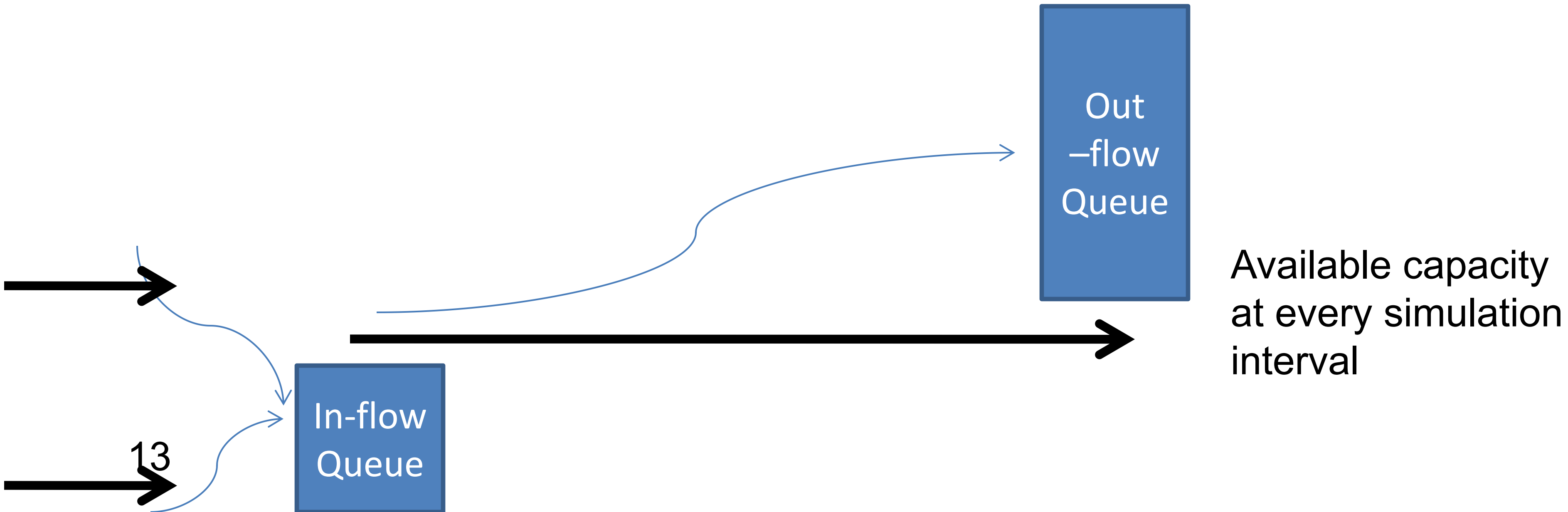
– CTM's variables are density, incoming and outgoing flows and speed

– Newell's variables are incoming and outgoing cumulative flow counts, from which the density and flows can be derived.

  - We do not have direct speed measurements and link travel times are calculated from vehicle trajectories (with arrival times at upstream node and departure time at downstream node)

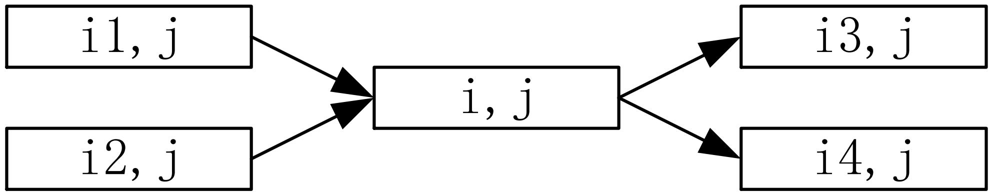
# Simulation Logic as Simple Queue

- Can be viewed as pseudo event-based simulation and we do not simulate how a vehicle moves inside the link
- Vehicle is moved into an in-flow queue at time $t_a$
- Calculate time entering the out-flow queue as $t_a$+FFTT (free-flow travel time)
- If the current simulation time t equals to or is later than $t_a$+FFTT, if the link out capacity is still available, move this vehicle to the next link, otherwise stay in the out-flow queue



# Network Traffic Flow Model

- Goal: define boundary conditions for each link
  - Downstream link capacity distribution at merges
  - Outflow distribution at diverges

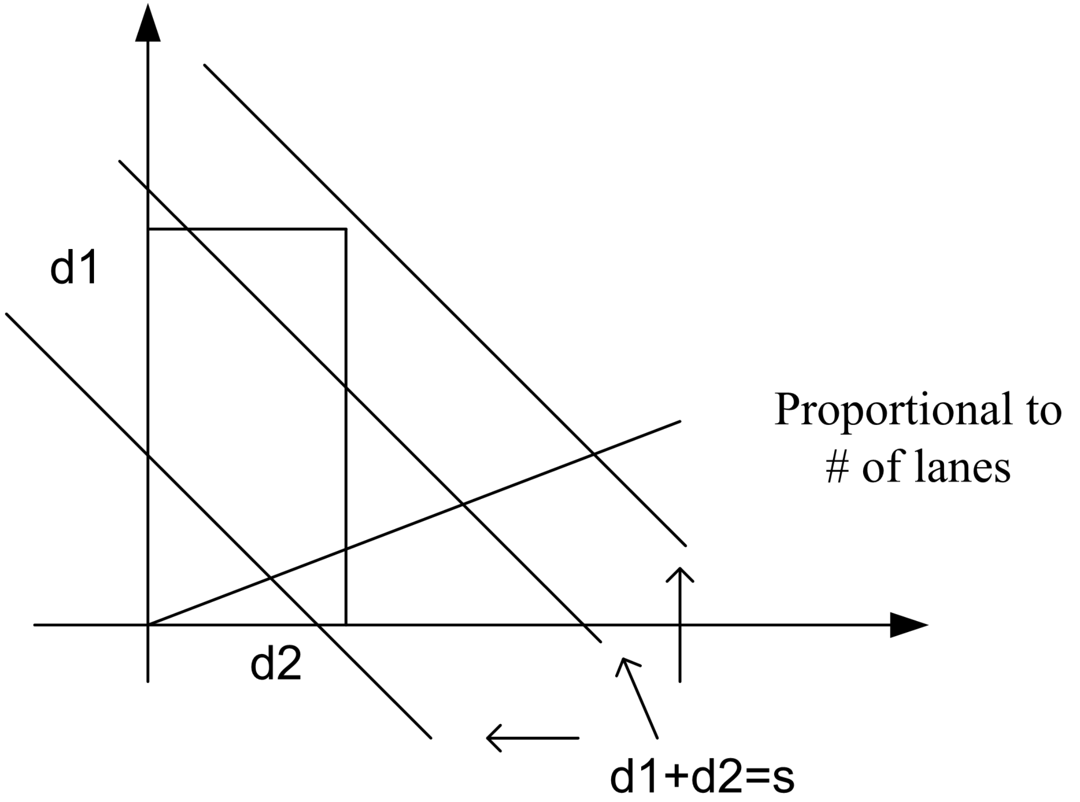



- Congestion from downstream link:

  $q(i1 \rightarrow i)+q(i2 \rightarrow i) <= \mathbf{w} \times (\mathbf{k_{jam}} - \mathbf{k_{i,j}(t)}) \times \mathbf{\Delta x}$

  or d1+d2<s
- Outflow distribution: $q(i \rightarrow i3)+q(i \rightarrow i4)=q(i)$
  - OD flow pattern, turning percentages

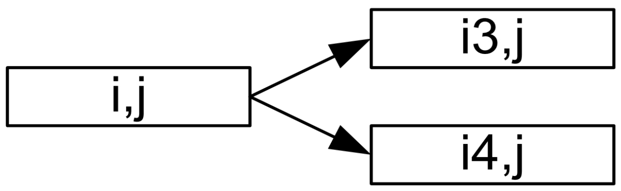
# In-Flow Capacity Distribution Schemes at Merges with two ramps or two freeways




Flows are assumed to be proportional to # of lanes whenever supplies are not adequate,

# Traffic Flow Model at Diverge

- As a meso-scopic model, DTALite moves vehicles with OD and path specific information instead of continuous flow
- Outflow distribution is dependent on
  - OD flow pattern, turning percentages



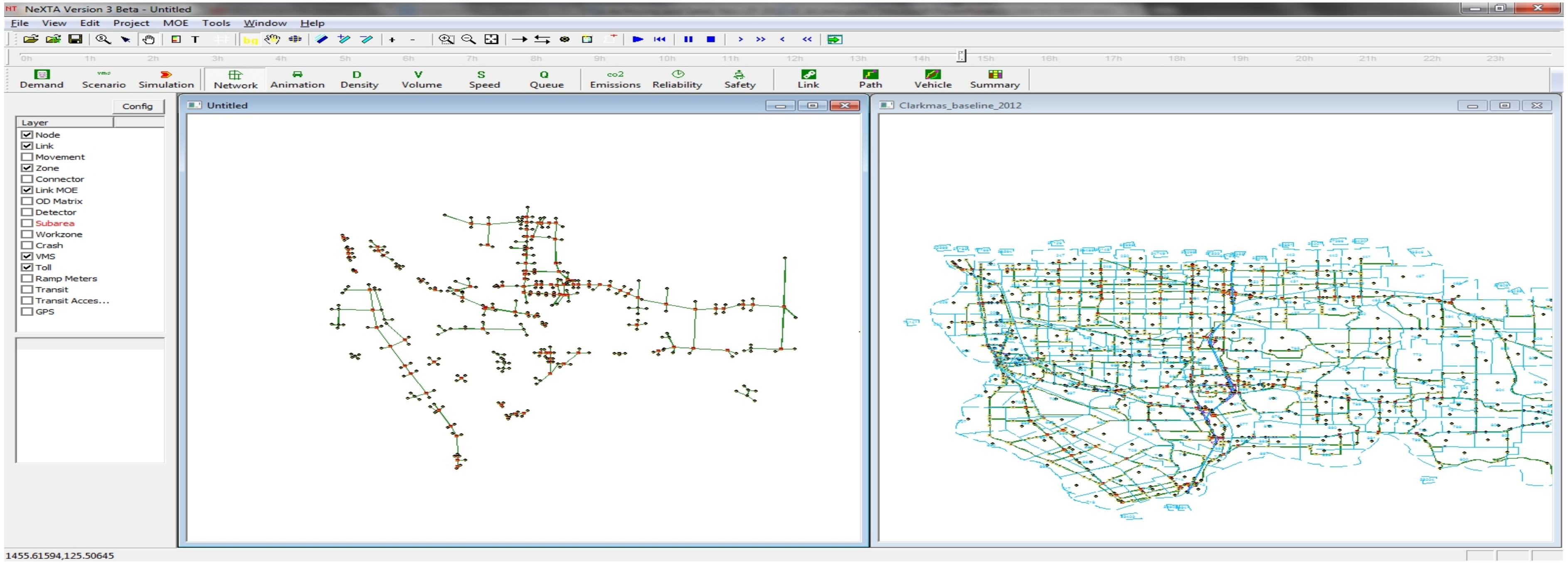
# Time-Dependent Shortest Path Algorithm

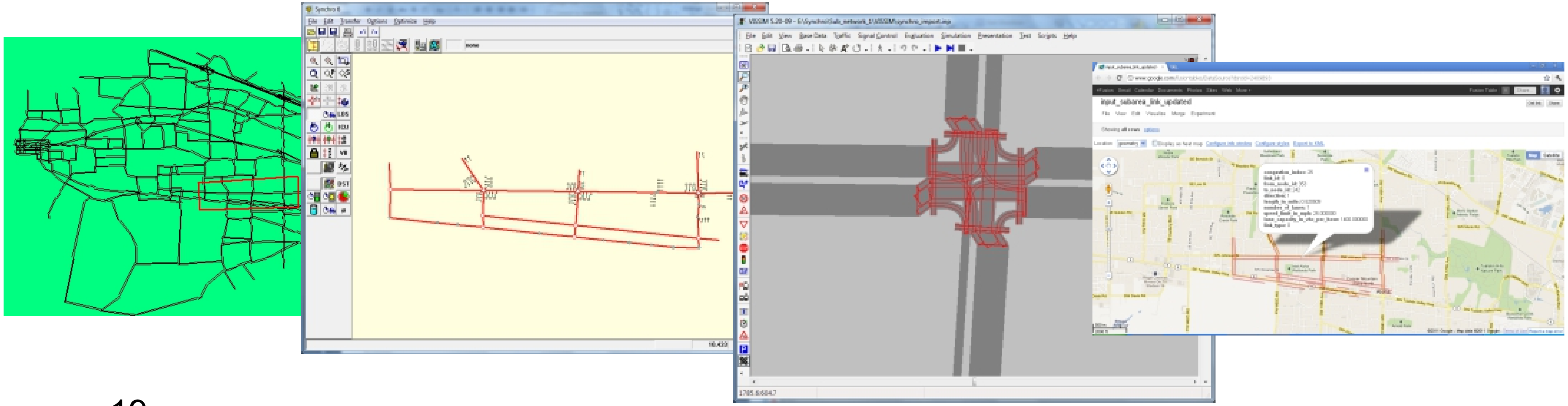
- C++ implementation
    - Use standard template library (STL) for complicating data structure (e.g. multi-dimensional vector, list, map (hash table)
    - Customized efficient structure for shortest path algorithm
- Multiple processor-oriented
  - OpenMP technique for using multiple processors
- Shortest path algorithm
  - Label correcting with deque
  - Single departure time, origin-based.
    - The algorithm is called iteratively for each departure time to calculate shortest paths for all departure times

# What NeXTA/DTALite Can Do

1. Open-source traffic simulation/modeling Data Hub
   - Integrated simulated and measured data management tools
   - Connection with Travel demand model, signal optimization tool (Synchro)
2. Large-scale dynamic traffic assignment
   - Typical network size: 2K~5K zones, 20K~50K links, 2~4 Million vehicles

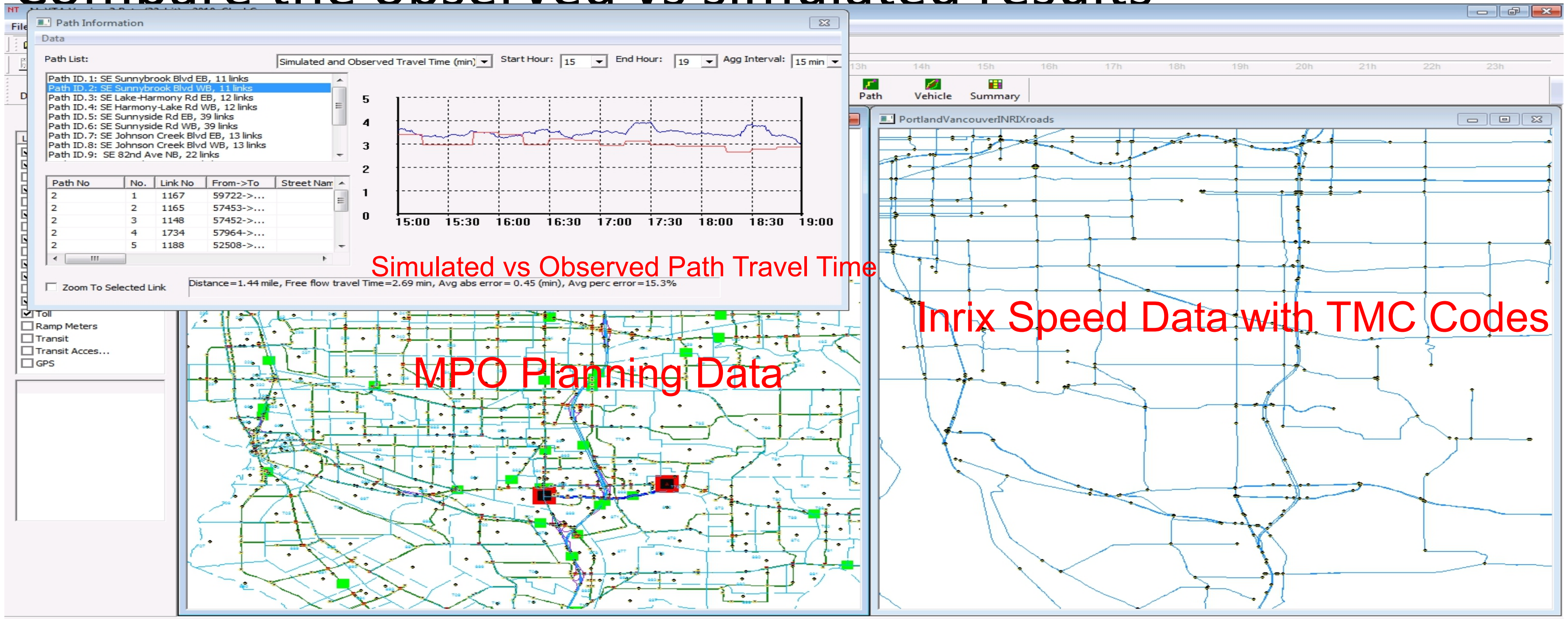
3. Network scenario analysis
   - Rich outputs: various operational performance measures
   - Road pricing application: Consider time-dependent toll, Heterogeneous Value of times


NeXTA Version 3 Beta - Untitled
File View Edit Project MOE Tools Window Help
Demand
Scenario
Simulation
Network
Animation
Density
Volume
Speed
Queue
Emissions
Reliability
Safety
Link
Path
Vehicle
Summary
Config
Layer
Node
Link
Movement
Zone
Connector
Link MOE
OD Matrix
Detector
Subarea
Workzone
Crash
VMS
Toll
Ramp Meters
Transit
Transit Acces...
GPS
Untitled
Clarkmas_baseline_2012
1455.61594,125.50645



# Connection with Traffic Sensor Data

1. Import TMC GIS data directly to NEXTA
2. Match TMC links with planning links using two layers
3. Fetch speed data from Inrix layer to AMS data bub
4. Compare the observed vs simulated results



# 1.1 Project Objective and Motivations

**Project Objective**

Enhance DTALite to provide a rigorous and computationally efficient tool to evaluate road pricing and crash-reduction strategies

The enhanced tool will enable practitioners to:

- Assess corridor and network-wide effects
- Conduct numerous alternatives analyses
- Evaluate recurring and non-recurring congestion
- Calculate performance measures that can be applied in investment-level planning and decision making

# 1.1 Project Objective and Motivations

## 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

# 1.2 What NeXTA/DTALite Can Do

1. Open-source traffic simulation/modeling Data Hub

   (1) Connection with signal optimization (Synchro) or microscopic simulation (VISSIM)

   (2) Integrated simulated and measured data management tools

2. Large-scale dynamic traffic assignment

   (1) Typical network size: 2000 zones, 20K links,1-2 Million vehicles

3. Network scenario analysis

   (1) Road pricing application: Consider time-dependent toll, Heterogeneous Value of times

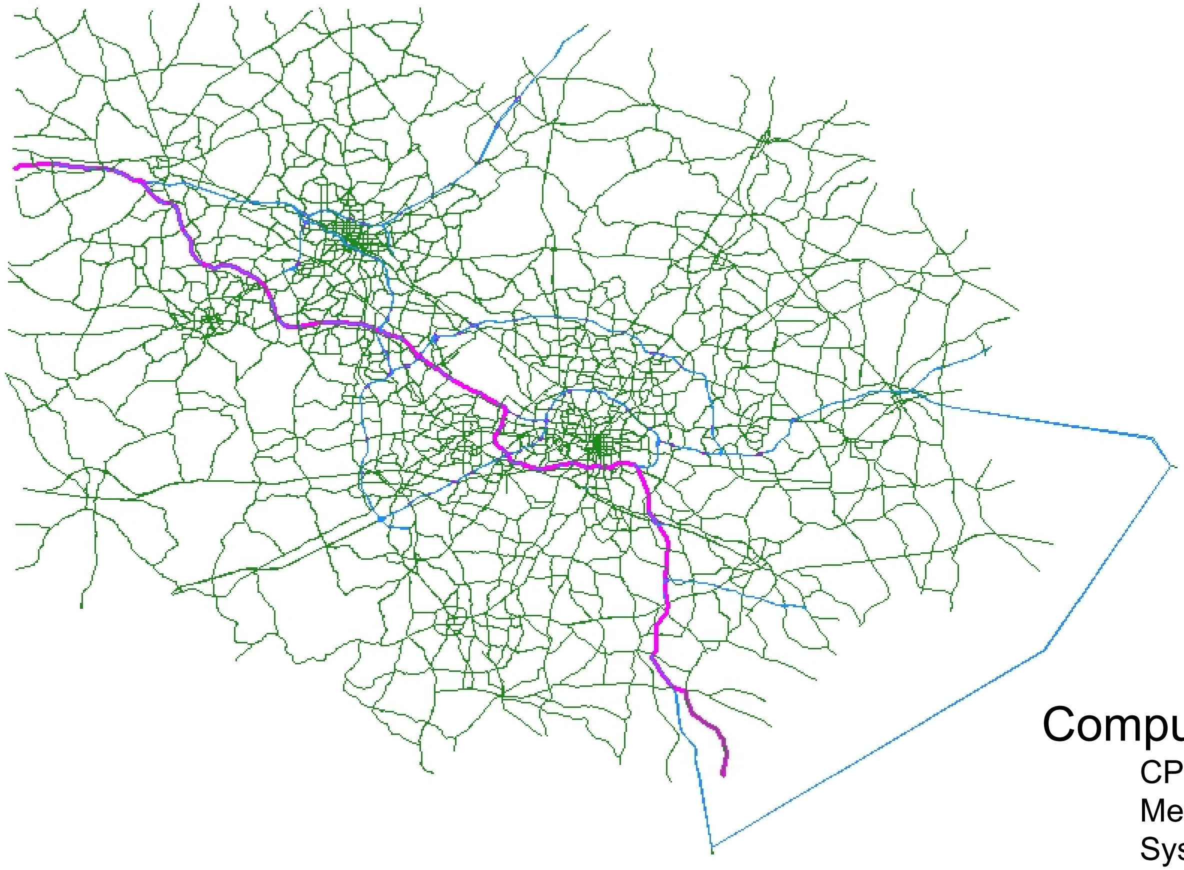
   (2) Emission study: Fast simulation for emission analysis (with MOVES Lite)

   (3) Safety planning: Predict annual crash rates based on link type and traffic volume

# 1.2 What NeXTA/DTALite Can Do

## 1) Fast Simulation

- A rigorous and computationally efficient tool to evaluate numerous alternatives analyses



**Network Statistics**

***Triangle Corridor Network (NC, USA)***

Zones = 2,389
Nodes = 9,528
Links = 20,258
Signals = 1,914
AM Trips = 1,064,703
Households = 490,000

2 min. 45 sec. / iteration
1hour for 20 iterations

Computer System Specification:

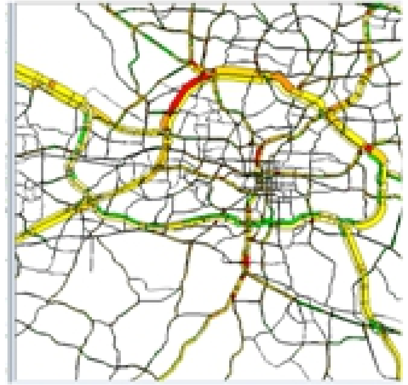
CPU: Intel i7-2960XM @ 2.70 GHz *8
Memory: 16.0 GB
System Type: 64-bit Windows 7

# 1.2 What NeXTA/DTALite Can Do

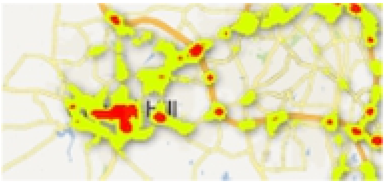
## 2) Rich Output

**NeXTA**

Large-scale Dynamic Traffic Assignment & Simulator

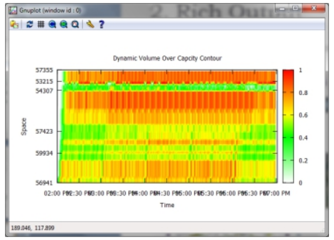


**Google Earth/Maps**

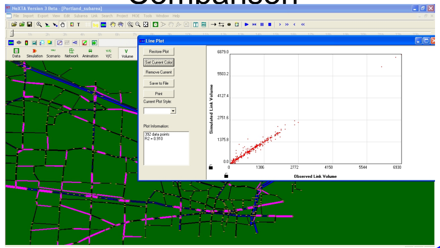


**Other Visualization**

Corridor-level VOC contour



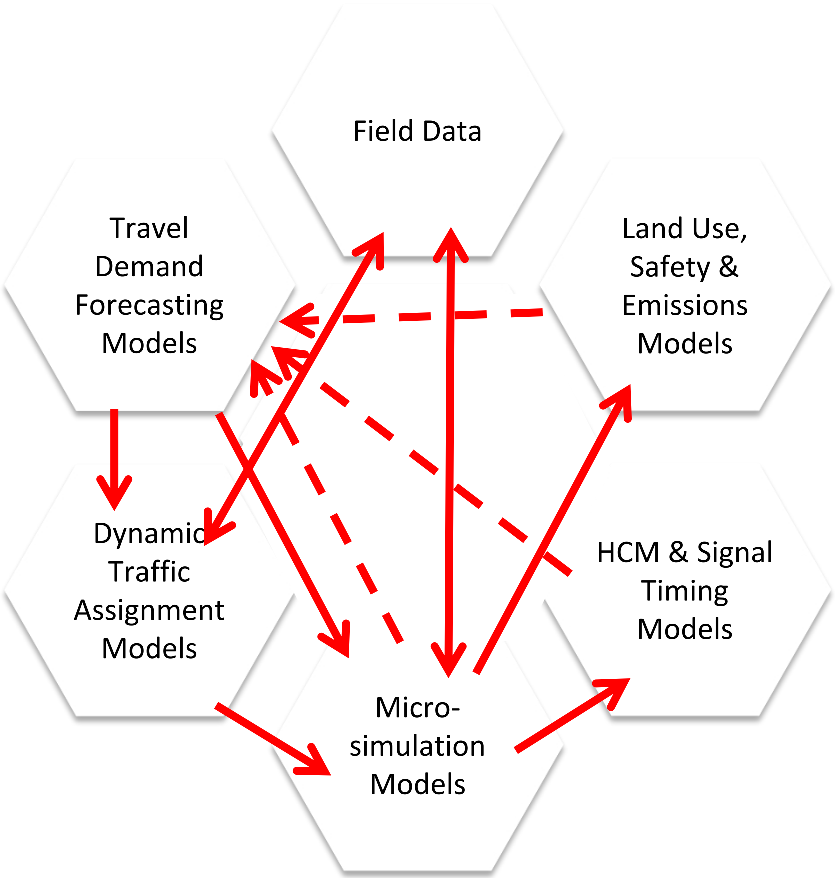

Simulation vs. Observation Comparison



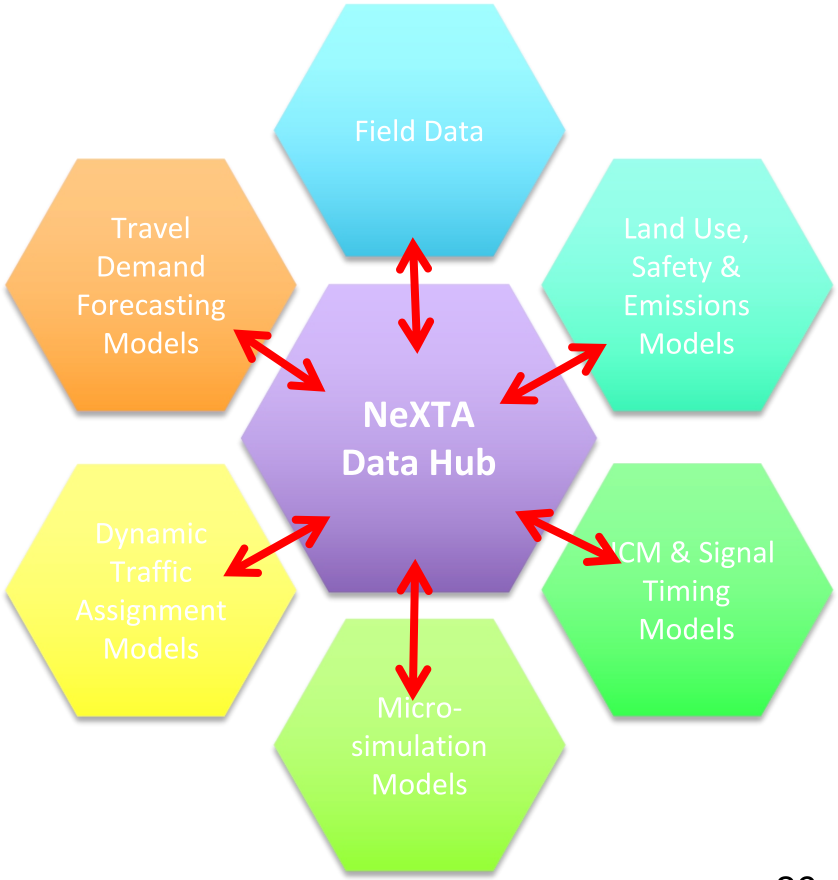
# 1.2 What NeXTA/DTALite Can Do

## 3) Integrated Modeling Practice

**Current Practice**
**Ad Hoc**

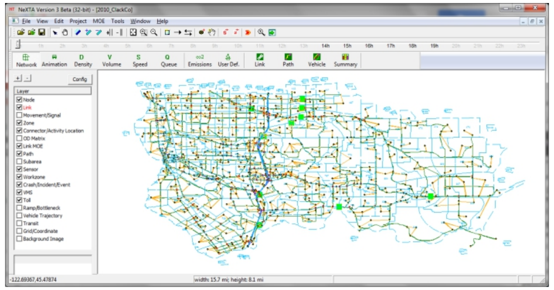


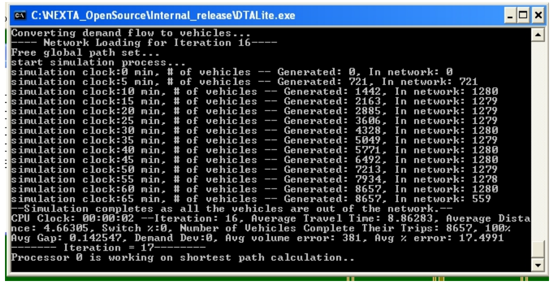

**With NeXTA Data Hub**
**Systematic**



# 1.3 Open-source Free Software Package

- **NEXTA: front-end GUI (C++)**
  – Version 2: GUI for TRANSIMS and DYNASMART
  – Version 3: GNU Open-source data hub
    - Import
      – Other regional planning models (TransCAD, VISSUM, Cube)
      – GIS shape files (household data without node layer)
      – Traffic volume, speed, GPS data, Google Public Transit Feed
    - Export
      – Google Earth, Google fusion tables
      – Prepare network and signal data for Synchro and VISSIM (through QEM)






- **DTALite: Open-source computational engine (C++)**
  – Light-weight and agent-based DTA
  – Built-in OD demand matrix estimation (ODME) program

# Useful Materials

- DTALite Software Release

  https://github.com/xzhou99/dtalite_software_release

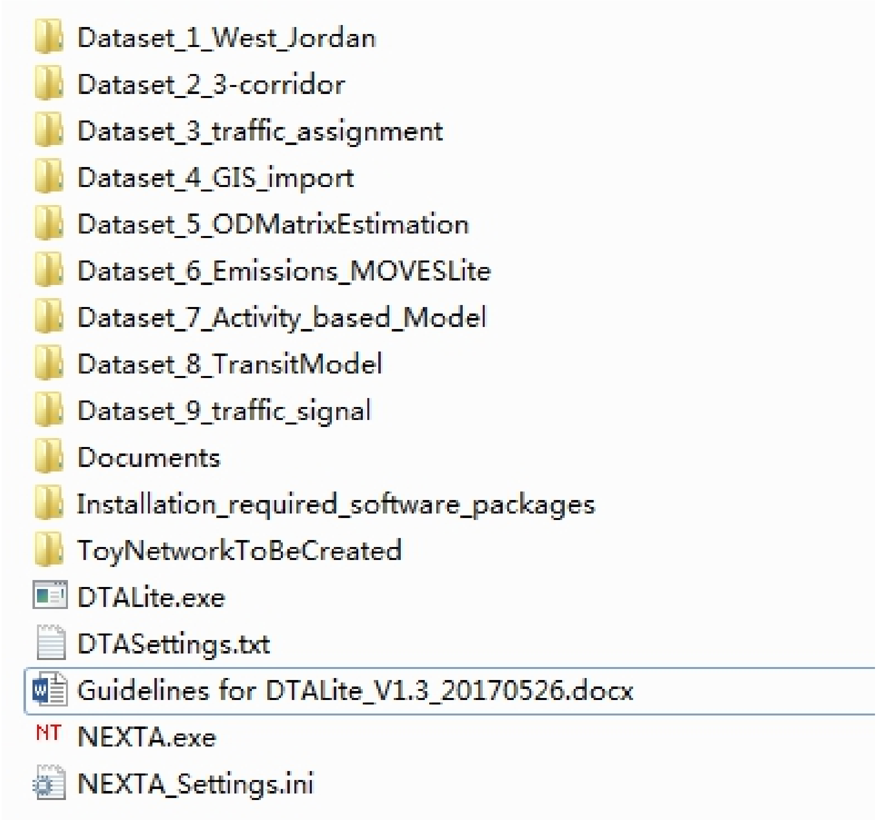
- NeXTA/DTALite Training Materials

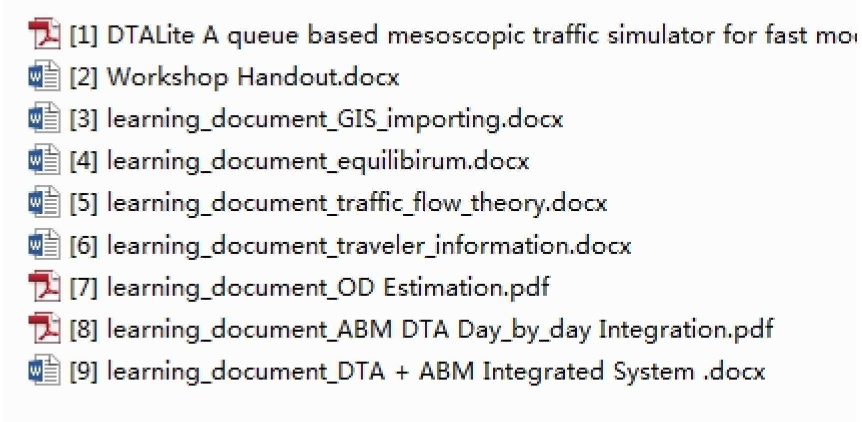
  https://github.com/xzhou99/learning-transportation/tree/master/lessons

- DTALite Source Codes
  - https://github.com/xzhou99/dtalite_beta_test

# Useful Materials

- Installation required software packages
- Datasets and Learning documents
- User Guide



# Module 2
# Introduction to DTA modelling principles

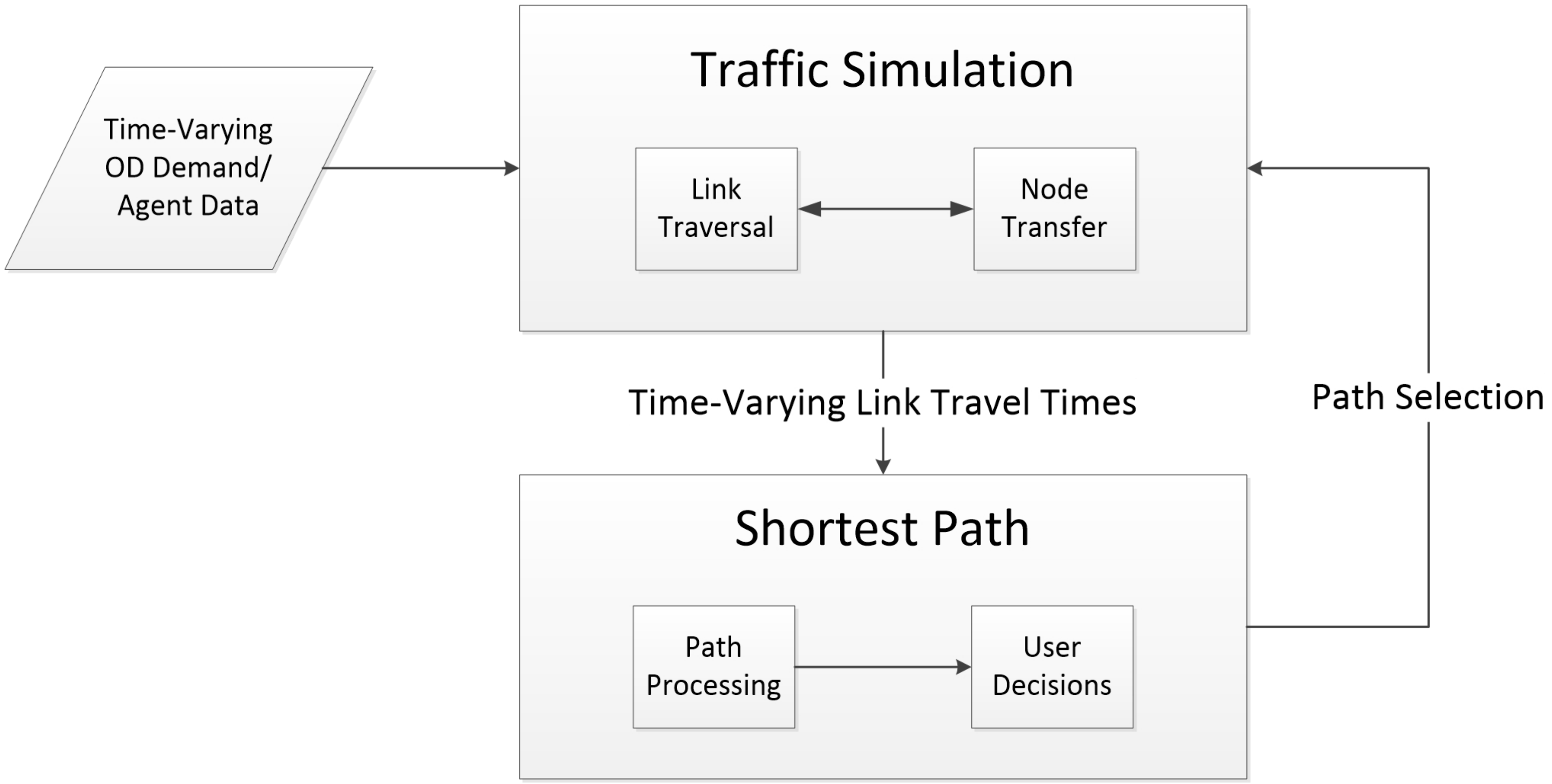
2.1 Dynamic Traffic Assignment Modelling Framework

2.2 Agent-based Routing

2.3 VOT Distribution

2.4 Multiple Traffic Simulation Models

# 2.1 DTA Modelling Framework



# 2.2 Agent-based Routing

**Common Assignment Methods**

(1) Traffic Analysis Zone (TAZ)-based

- From zone centroid to zone centroid
- The same value of time for each type of vehicles

(2) Agent-based

- From activity location to activity location
- Each vehicle has its own value of time, value of information
- Important for road tolling analysis, traveler information provision study

# 2.2 Agent-based Routing

- Routing performed for each individual agent
- Each agent has multiple dimensions of travel decisions
  - Origin, destination, departure time, path
  - Demand class (LOV, HOV, truck) or (HBW, HBO, NHB)
  - Information class (Historical, Pre-trip, En-route)

# 2.2 Agent-based Routing

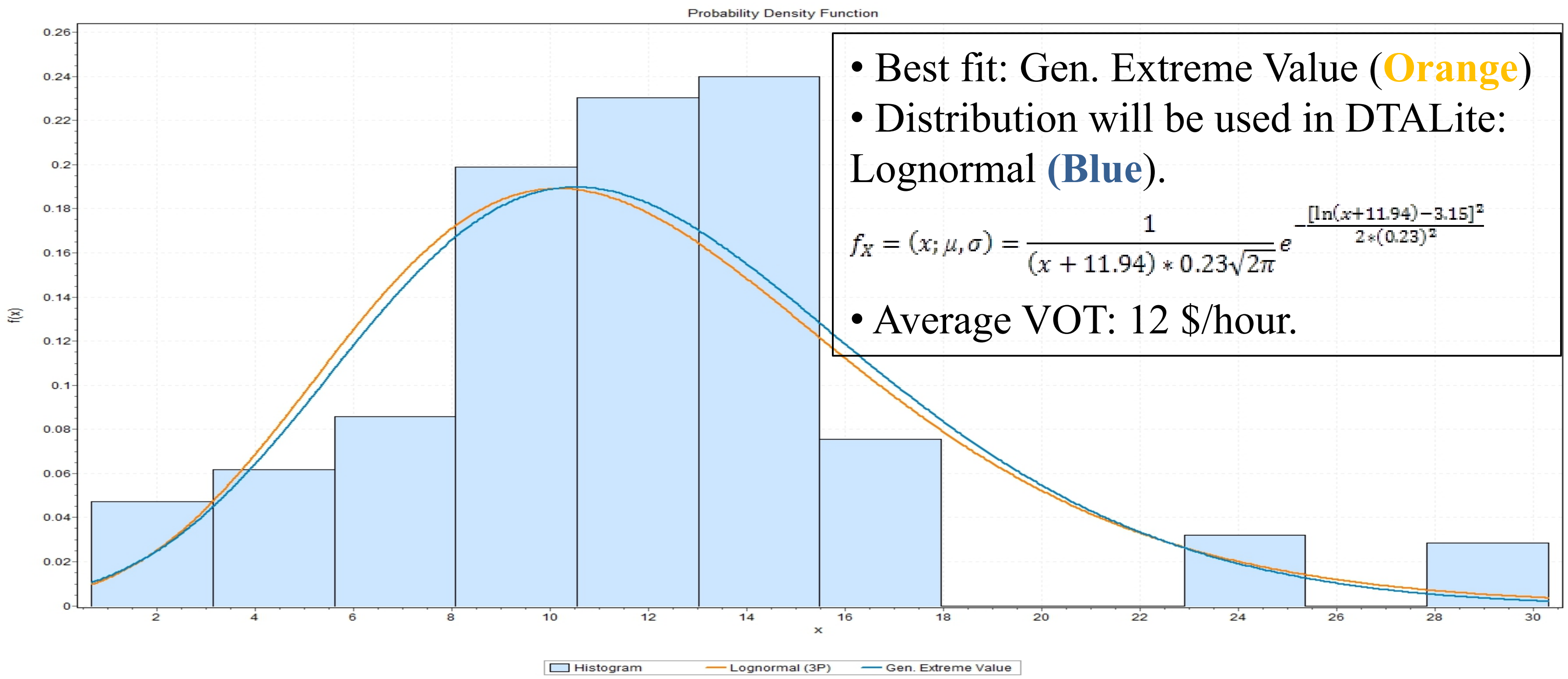
- Individual generalized cost function

$$Cost = Travel\,Time * VOT + Toll$$

  - Can consider multiple factors
    - Value of time, Value of reliability, Value of safety
  - Perform routing algorithm individually for each vehicle/agent
  - Can adjust origin/destination/departure time/path at each iteration (day)

# 2.3 VOT Distribution

- Assumption:
  - VOT : 50% of hourly rate (Concas and Kolpakov, 2009)




- Best fit: Gen. Extreme Value (**Orange**)
- Distribution will be used in DTALite: Lognormal **(Blue)**.

$$f_X = (x;\mu,\sigma) = \frac{1}{(x+11.94)*0.23\sqrt{2\pi}} e^{-\frac{[\ln(x+11.94)-3.15]^2}{2*(0.23)^2}}$$

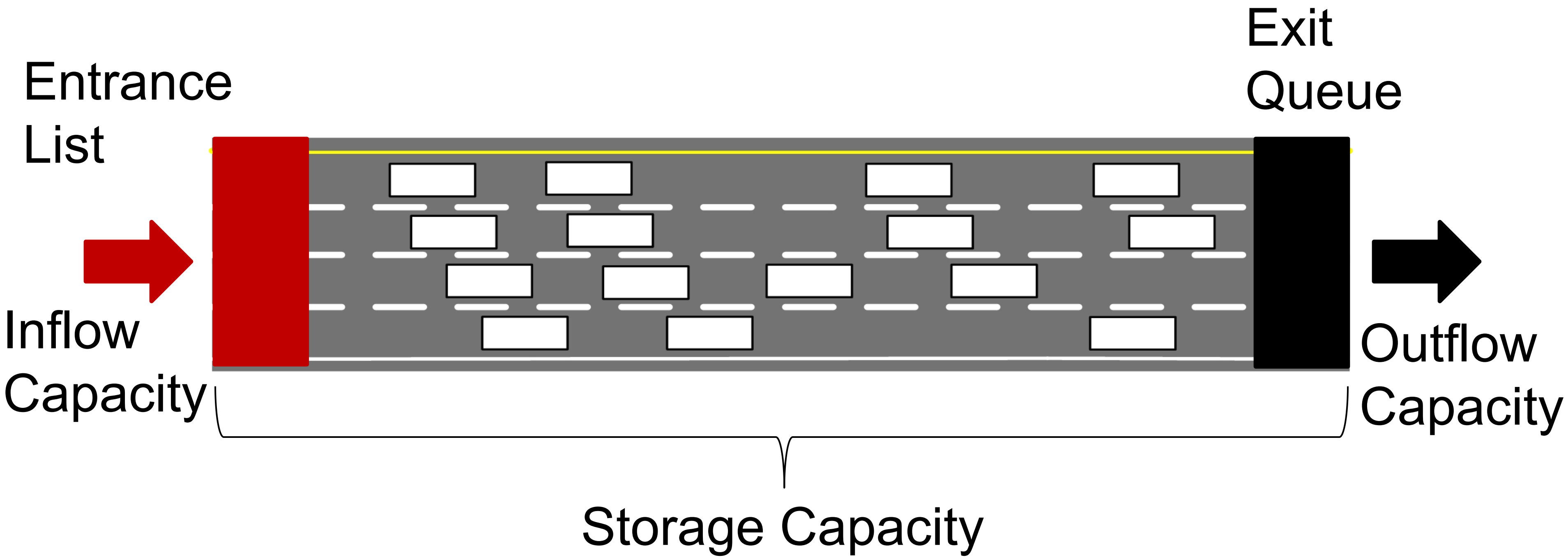
- Average VOT: 12 $/hour.

# 2.4 Multiple Traffic Simulation Models

- BPR travel time functions
- Point queue (relaxed storage constraints)
- Spatial queue (similar to DYNASMART-P model)
- Newell's model (similar to Cell Transmission Model)
  - Shockwave propagation

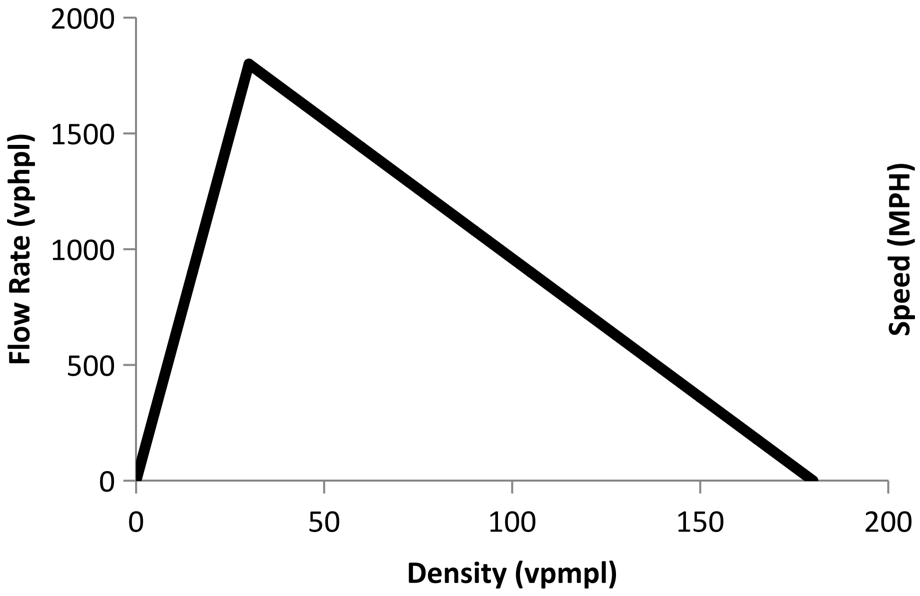
# 2.4 Multiple Traffic Simulation Models

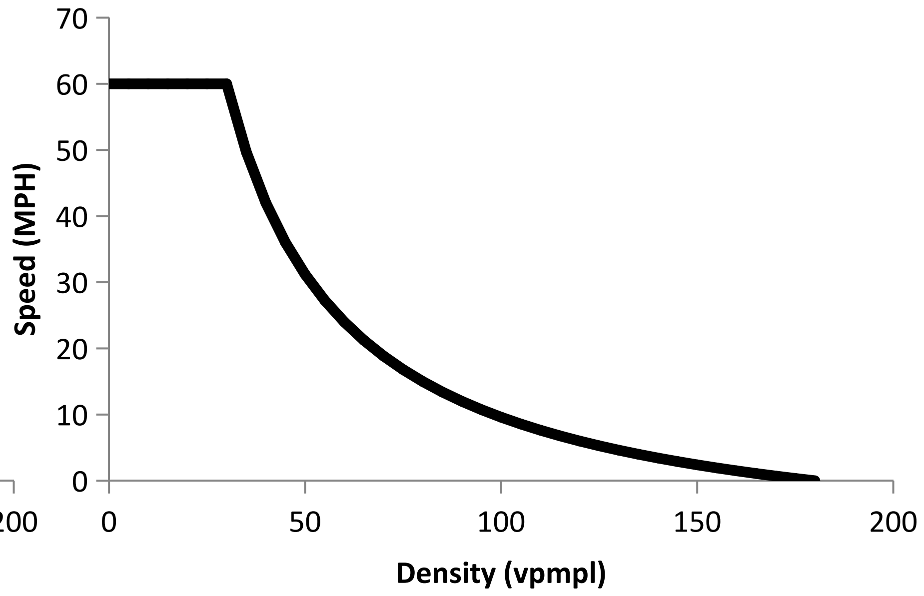
- **Newell's simplified kinematic wave model**
  - Outflow capacity
  - Inflow capacity
  - Storage capacity



# 2.4 Multiple Traffic Simulation Models

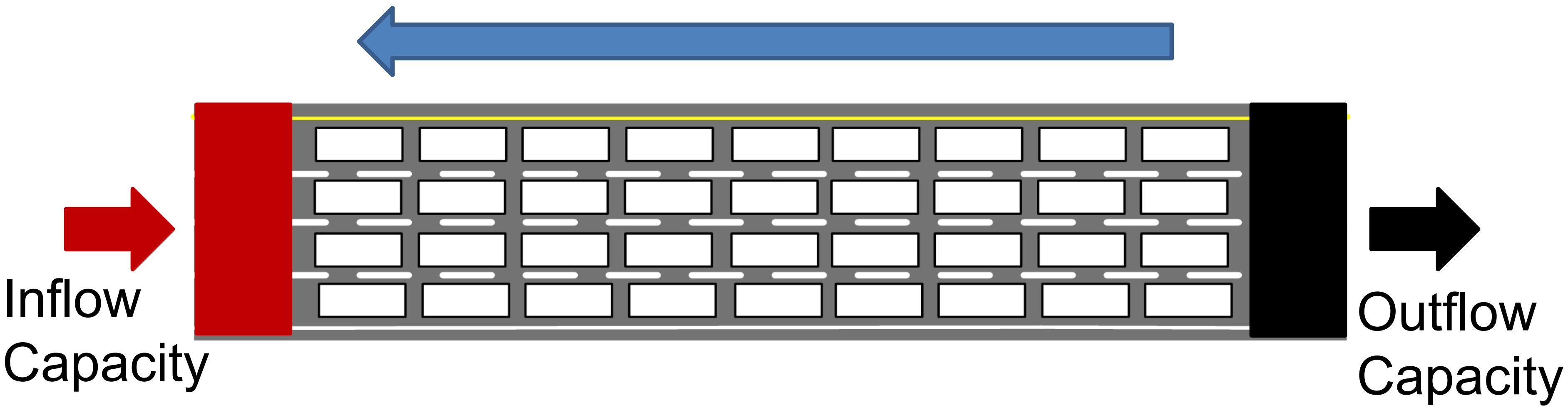
- **Newell's simplified kinematic wave model**
  - Triangular flow-density relationship
  - Free flow speed, jam density, backward wave speed



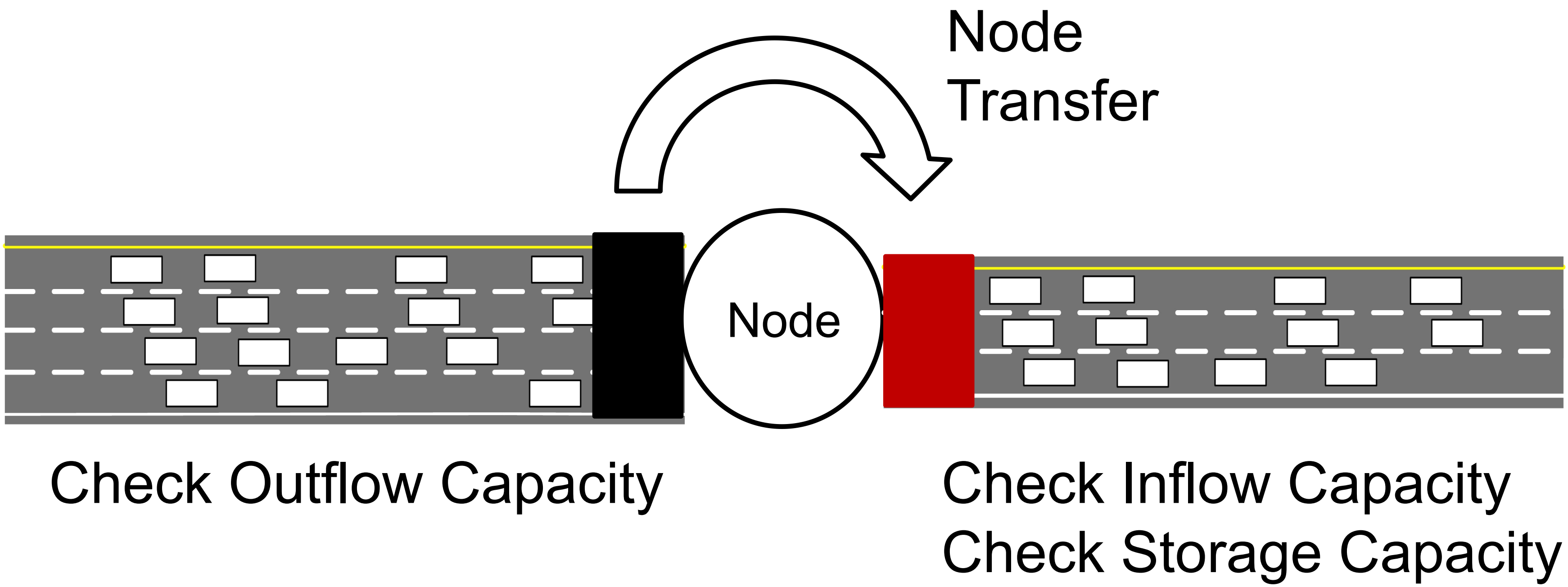
# 2.4 MultipleTraffic Simulation Models

- **Queue propagation**
  - Inflow capacity = outflow capacity



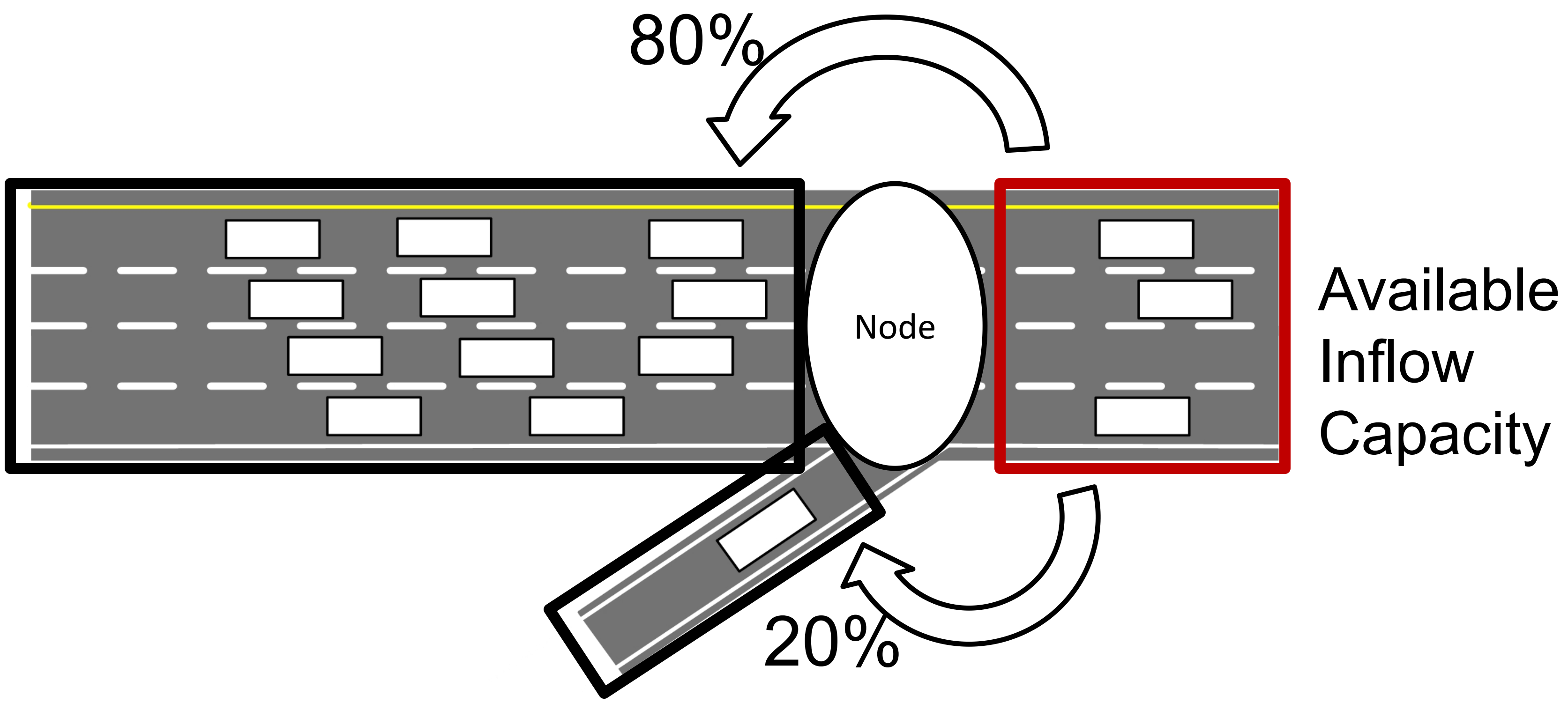
# 2.4 Multiple Traffic Simulation Models

- **Node transfer**



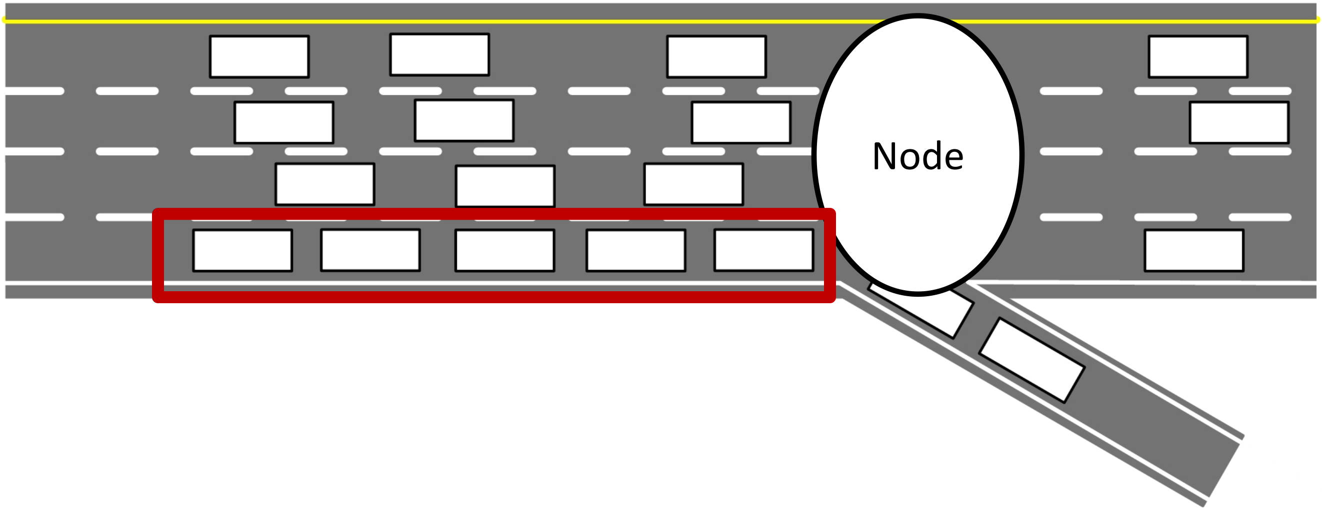
# 2.4 Multiple Traffic Simulation Models

- **Merge Models**
  - Distribute inflow capacity to upstream links
  - Lane & demand-based methods



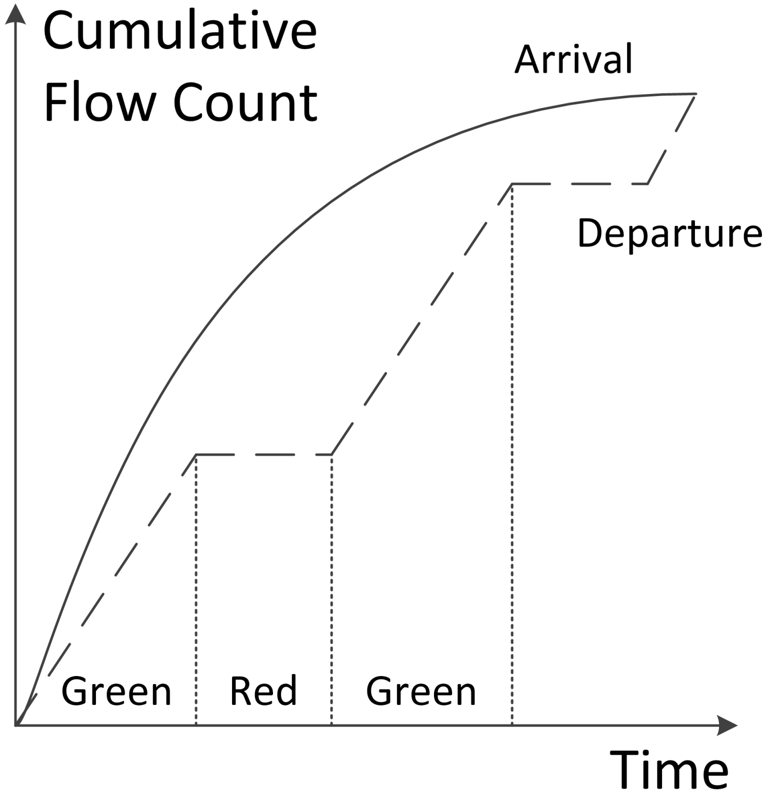
# 2.4 Multiple Traffic Simulation Models

- **Diverge Models**
  - Different conditions by lane
  - First-In-First-Out (FIFO) constraint
  - Relaxation to prevent extreme bottlenecks



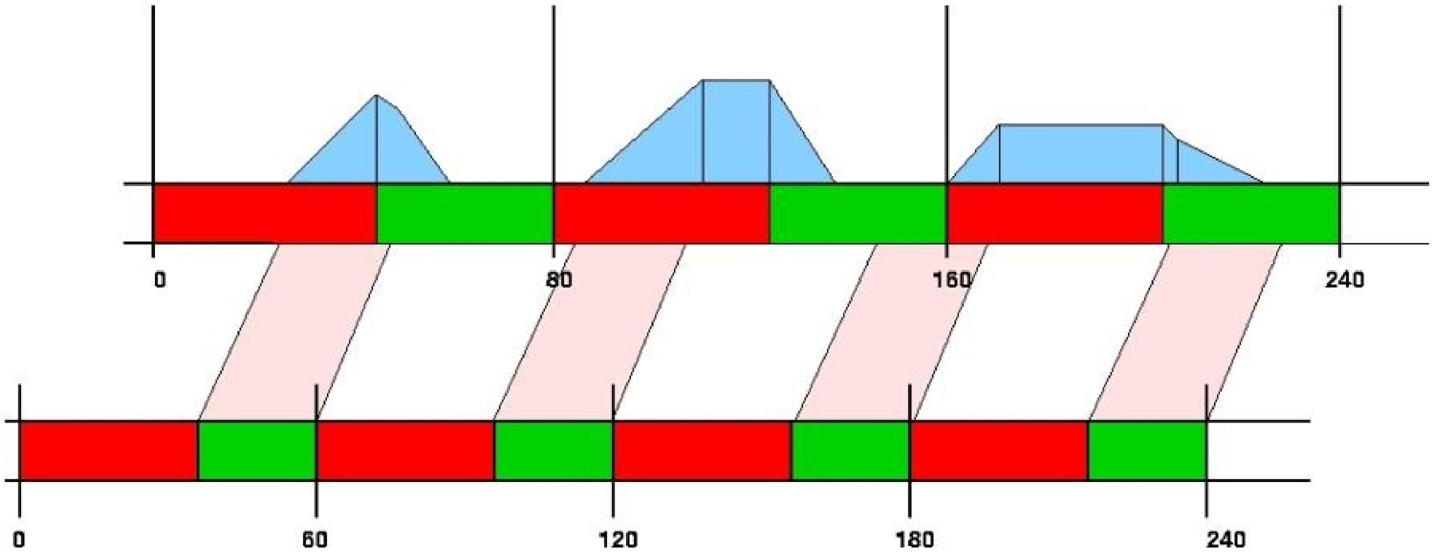
# 2.4 Multiple Traffic Simulation Models

- **Signalized Intersections**
  – Effective green time, saturation flow rate, movement-based capacity
  – Relaxed inflow constraints



# 2.4 Multiple Traffic Simulation Models

- **Signal Timing & Hourly Capacity**
  - Input: Average hourly capacity, cycle time, offset at node
  - Output: Effective green time per cycle(capacity/saturation flow rates)



# Module 3
# Visualization Features in NeXTA

3.1 Basic GUI features

3.2 View/Edit data files in NEXTA’s “project” menu

3.3 Integration with assignment model

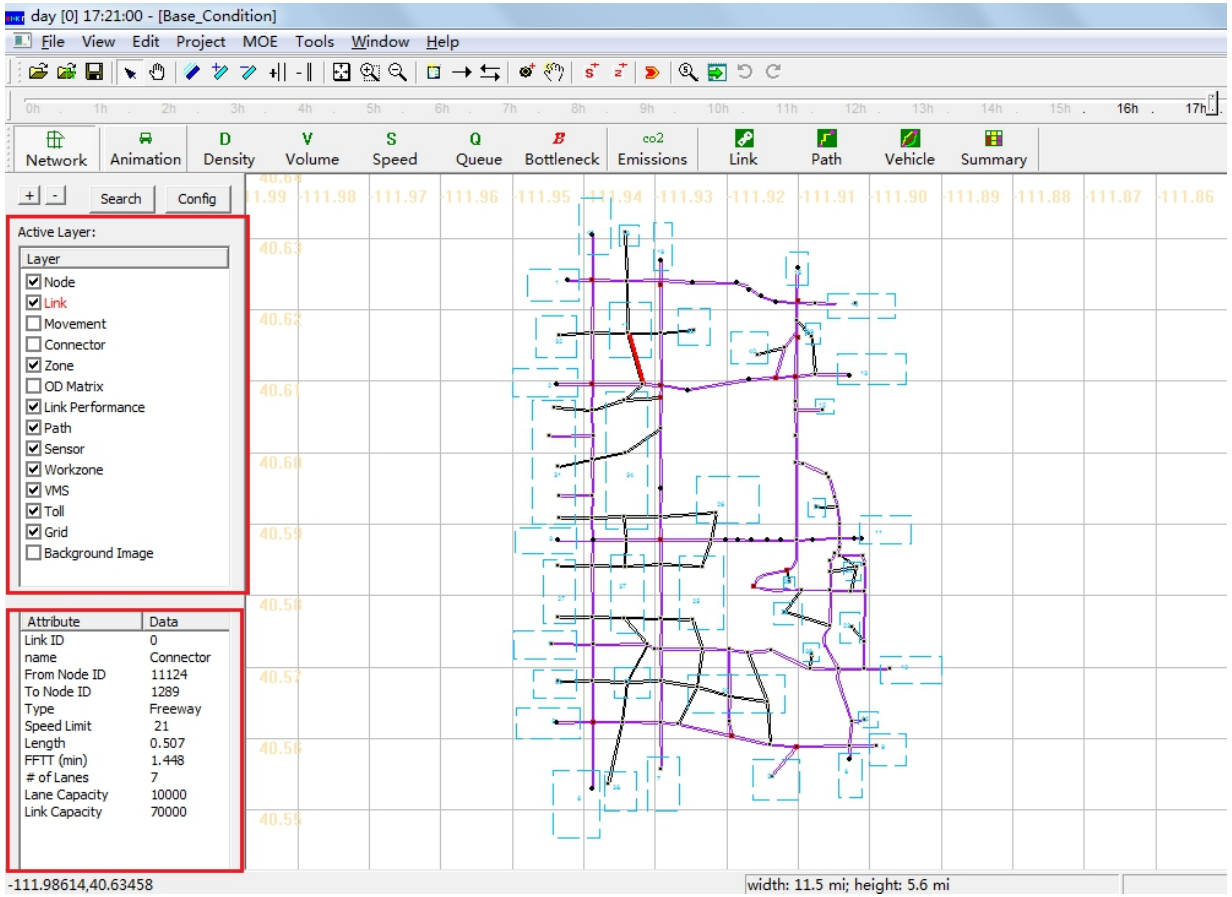
3.4 Advanced visualization functions of NEXTA

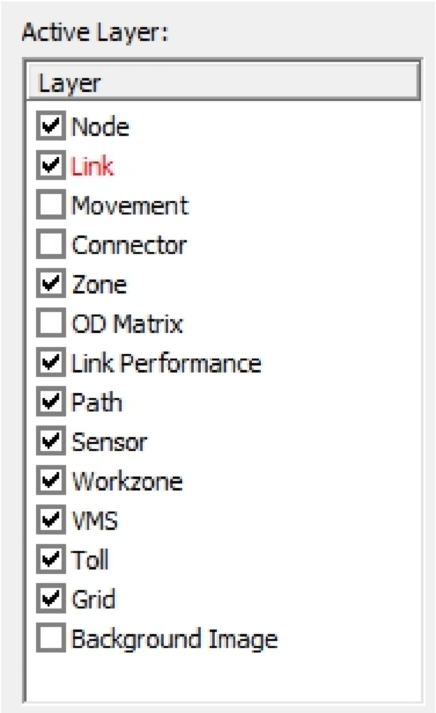
3.5 MOE related display

3.6 Data exporting to Google Earth /GIS package

# 3.1 Basic GUI features

- Turn on and off GIS layers; Move around, select node and links; Toolbars for editing networks
- Open project folder (CSV file format)



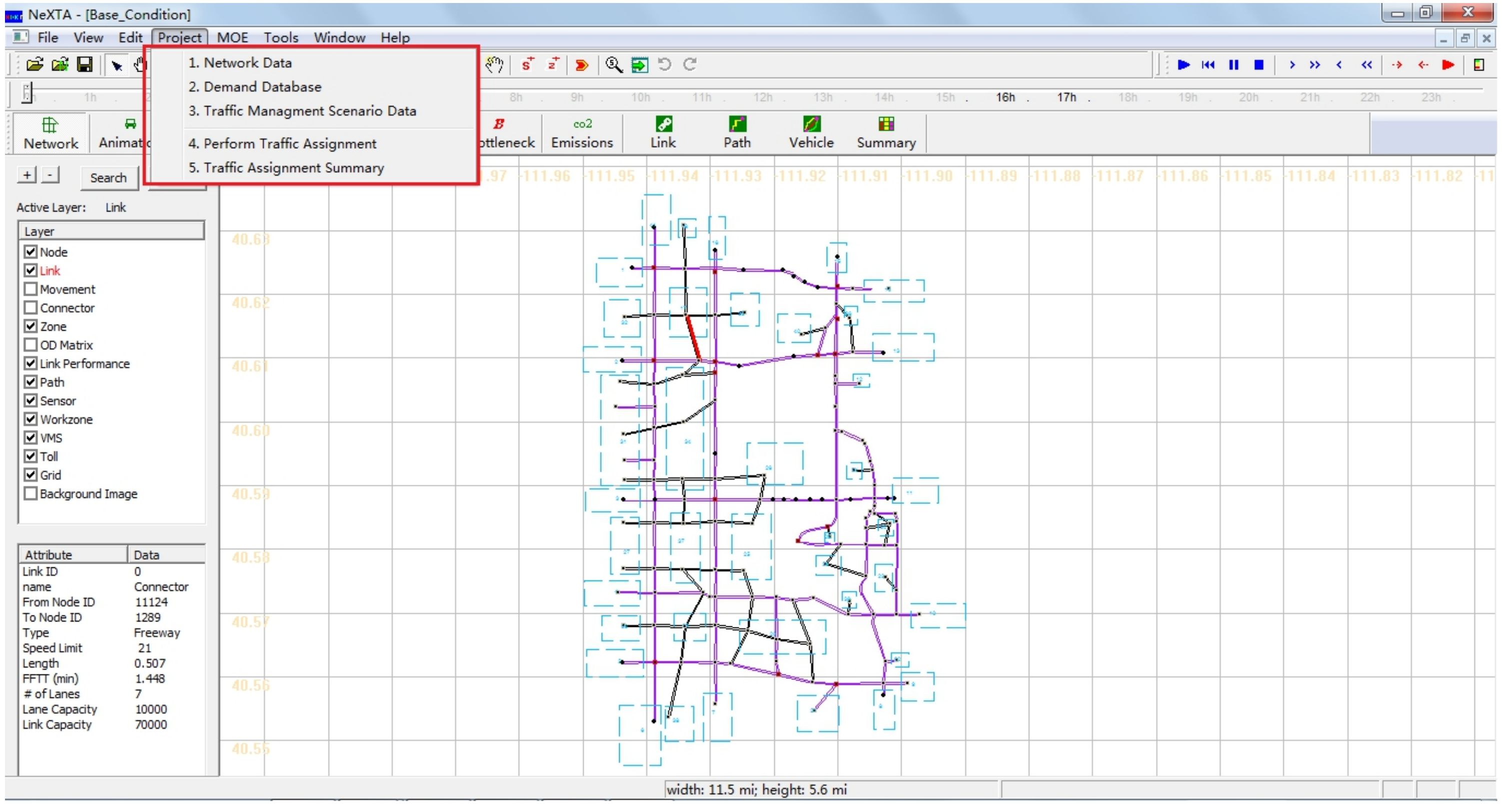
# 3.1 Basic GUI features

- Turn on and off GIS layers; Move around, select node and links; Toolbars for editing networks
- Open project folder (CSV file format)

| | | | |
|---|---|---|---|
| input_activity_location.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 1 KB |
| input_demand.csv | 1/11/2013 9:43 ... | Microsoft Excel ... | 24 KB |
| input_demand_file_list.csv | 1/23/2017 12:08... | Microsoft Excel ... | 2 KB |
| input_demand_type.csv | 6/30/2016 3:40 ... | Microsoft Excel ... | 1 KB |
| input_link.csv | 9/4/2017 3:53 PM | Microsoft Excel ... | 65 KB |
| input_link_type.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 1 KB |
| input_movement.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 25 KB |
| input_node.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 17 KB |
| input_node_control_type.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 1 KB |
| input_scenario_settings.csv | 5/25/2017 5:06 ... | Microsoft Excel ... | 1 KB |
| input_zone.csv | 1/22/2017 4:14 ... | Microsoft Excel ... | 11 KB |
| ODME_link_based_log.csv | 6/14/2017 10:54... | Microsoft Excel ... | 1 KB |
| ODME_zone_based_log.csv | 6/14/2017 10:54... | Microsoft Excel ... | 0 KB |
| optional_MOE_settings.csv | 3/22/2016 11:07... | Microsoft Excel ... | 1 KB |
| optional_vehicle_type.csv | 3/22/2016 10:54... | Microsoft Excel ... | 1 KB |
| output_agent.csv | 6/14/2017 10:54... | Microsoft Excel ... | 0 KB |
| output_day_to_day_MOE.csv | 6/14/2017 10:54... | Microsoft Excel ... | 9 KB |
| output_LinkMOE.csv | 6/14/2017 10:54... | Microsoft Excel ... | 0 KB |

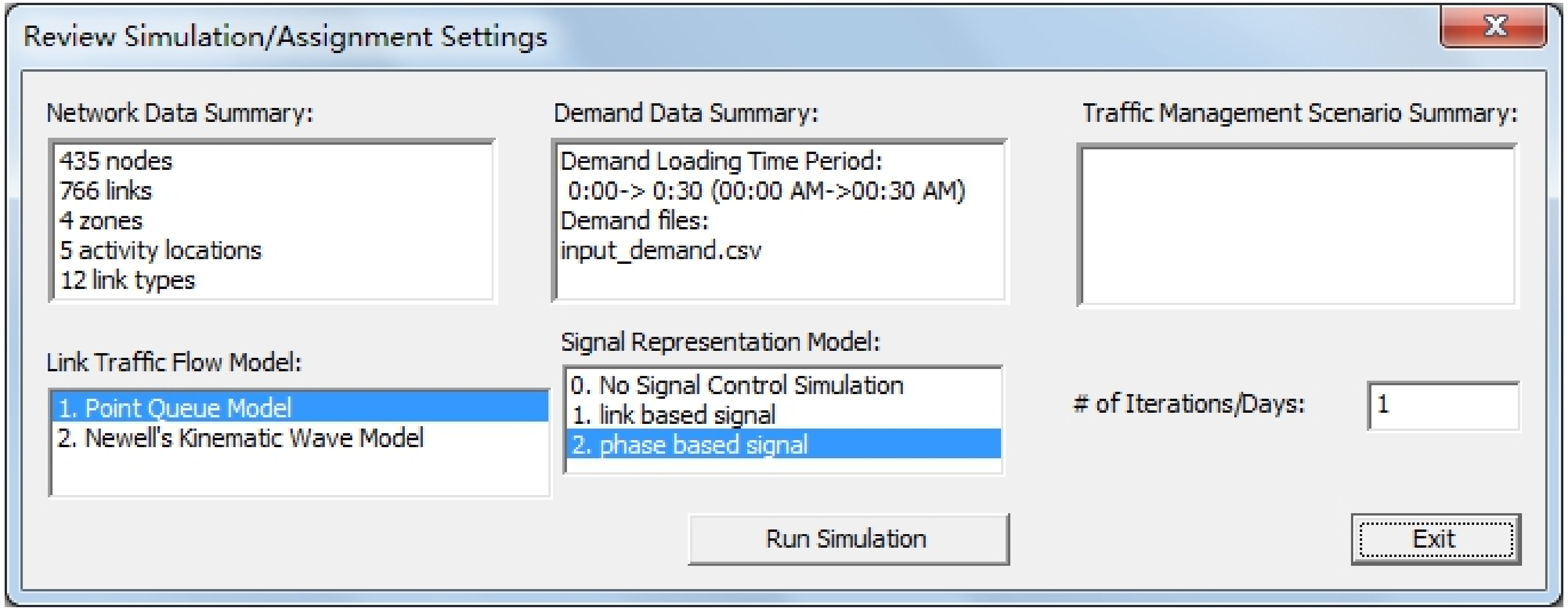
# 3.2 View/Edit data files in "project" menu

- Node/link/zone/activity location
- Demand meta database
- Scenario files



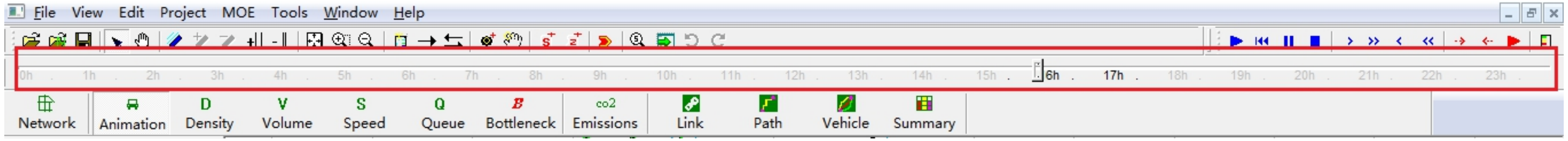
# 3.3 Integration with assignment model

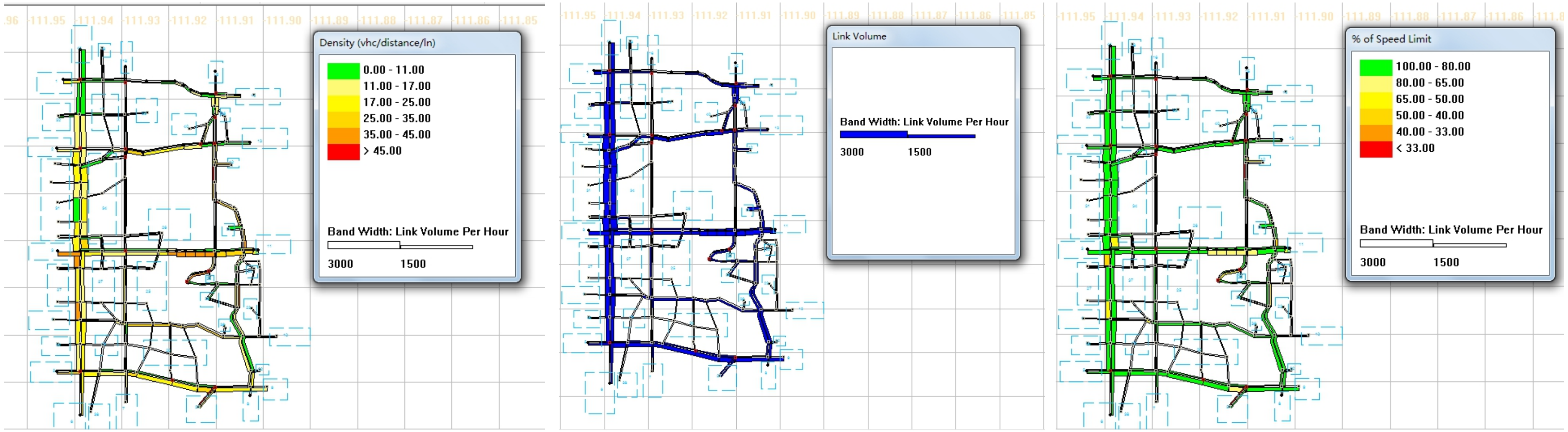
- Traffic flow model
- Traffic Assignment method



# 3.4 Advanced visualization functions of NeXTA

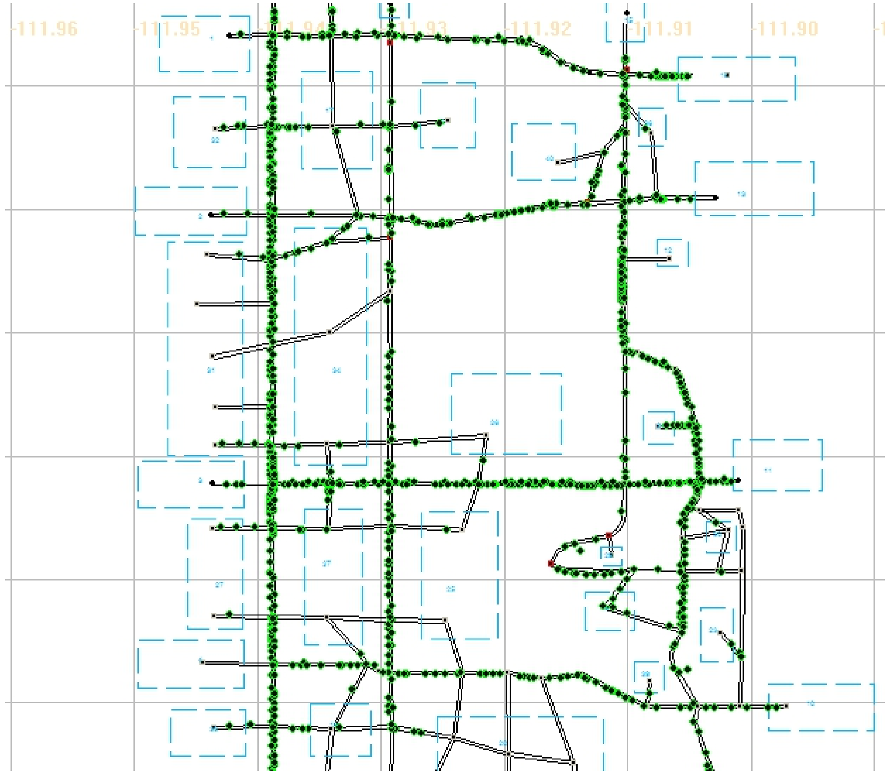
- 24-hour Time control/Clock bar
- Volume (bandwidth), density, speed
- Animation and queue: (turn off node layer and bandwidth)

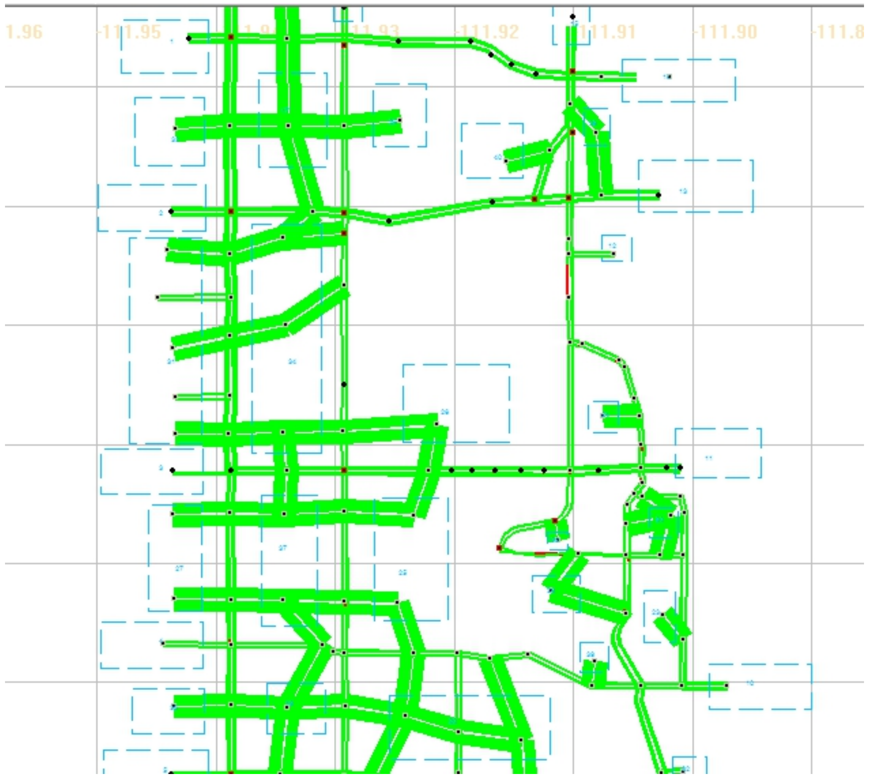


# 3.4 Advanced visualization functions of NeXTA

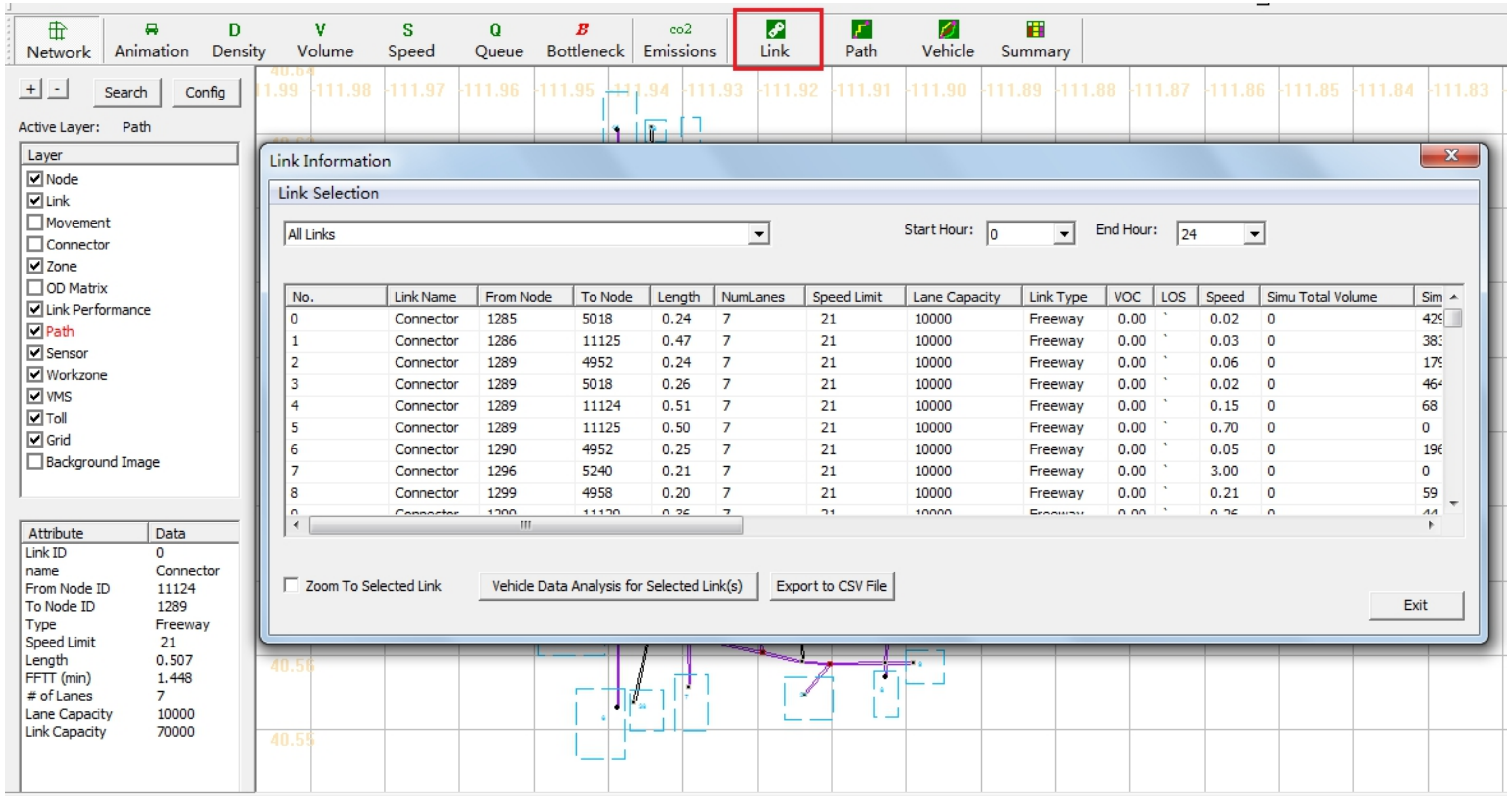
- 24-hour Time control/Clock bar
- Volume (bandwidth), density, speed
- Animation and queue: (turn off node layer and bandwidth)





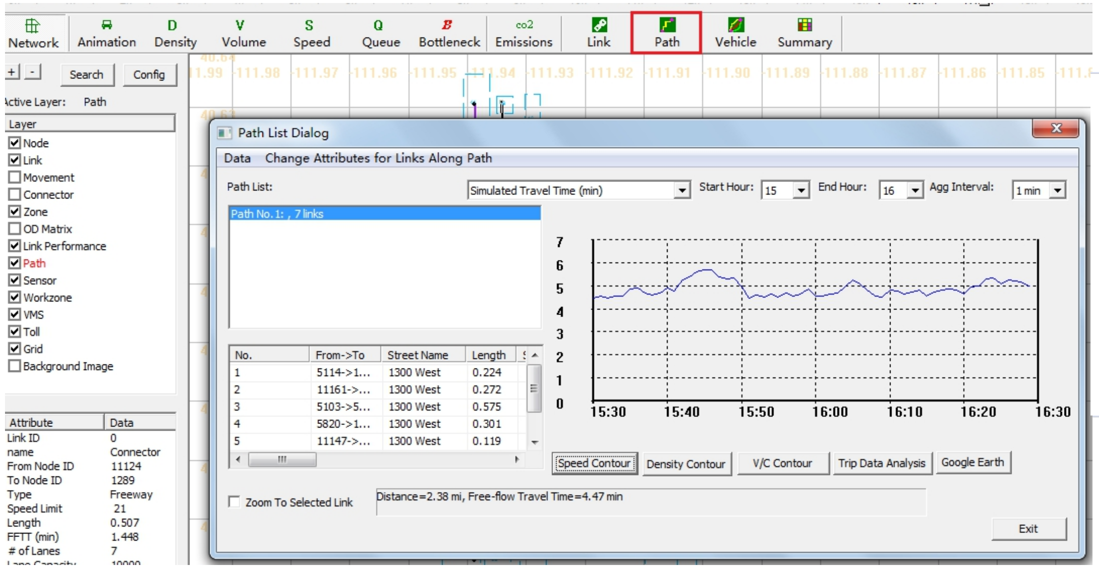
# 3.5 MOE related display

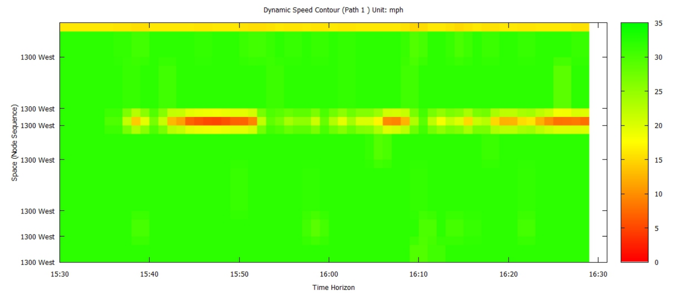
- Link-related display
- Path-related display
- Vehicle and Summary Charts

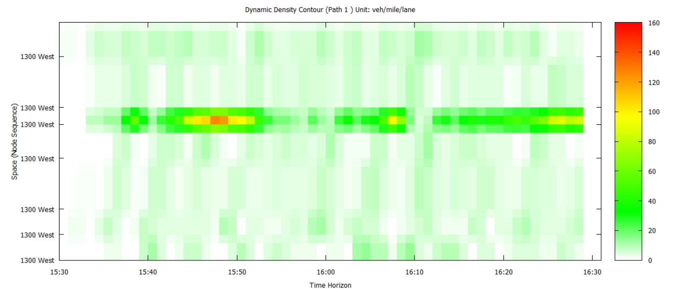


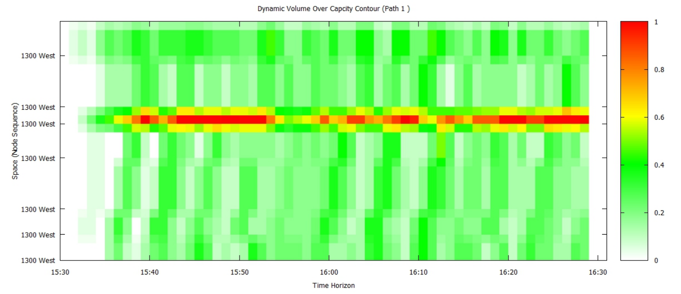
# 3.5 MOE related display

- Link-related display
- Path-related display
- Vehicle and Summary Charts



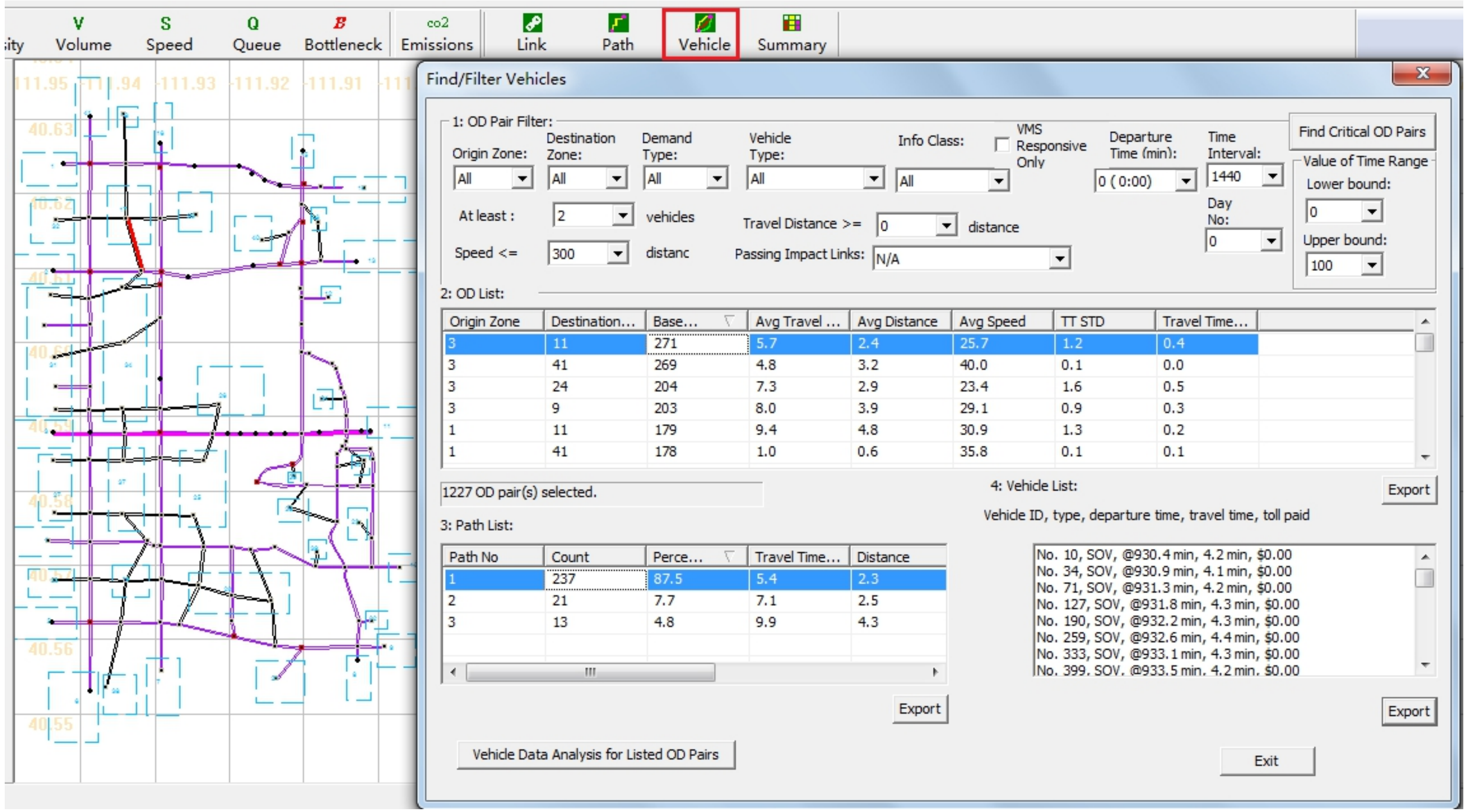



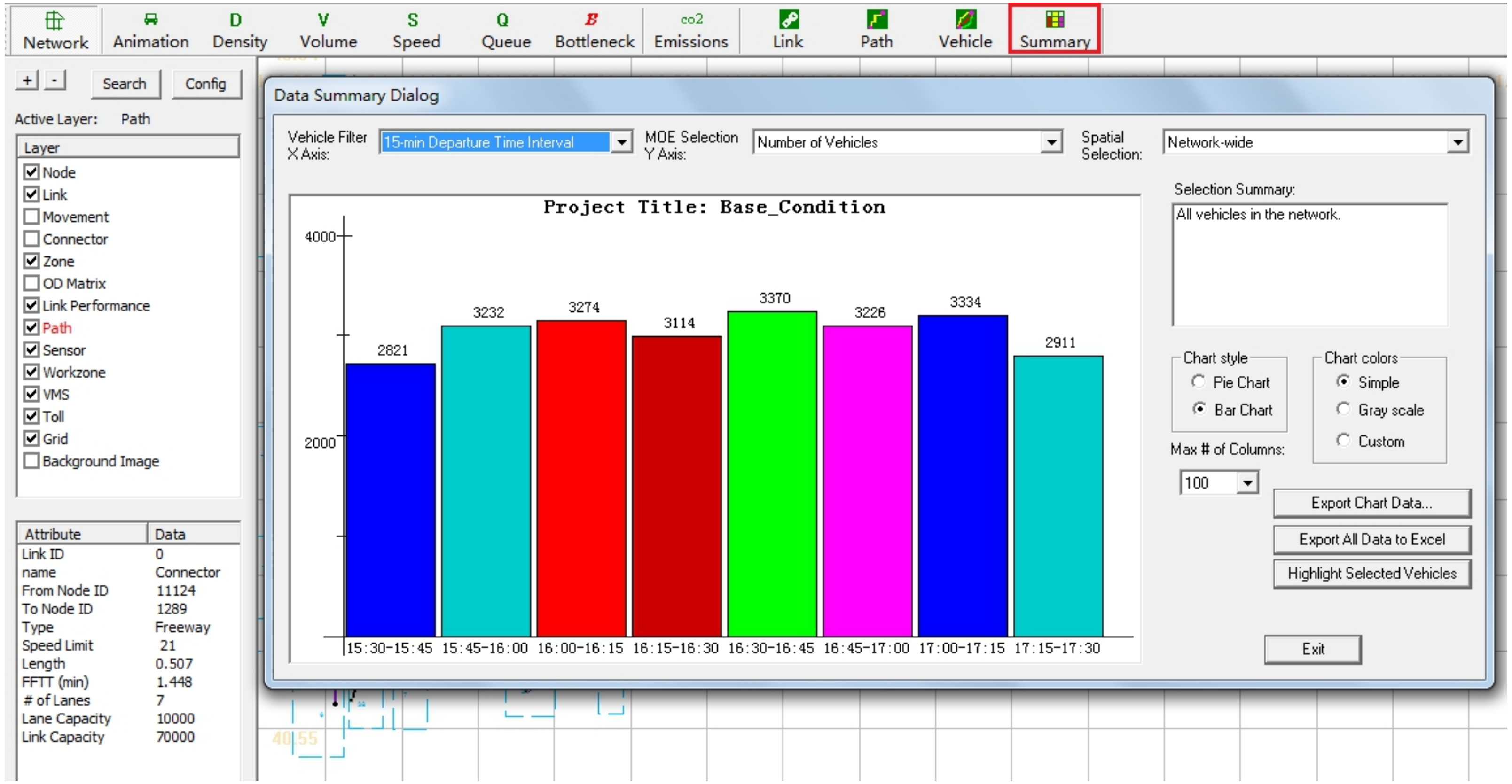
# 3.5 MOE related display

- Link-related display
- Path-related display
- Vehicle and Summary Charts



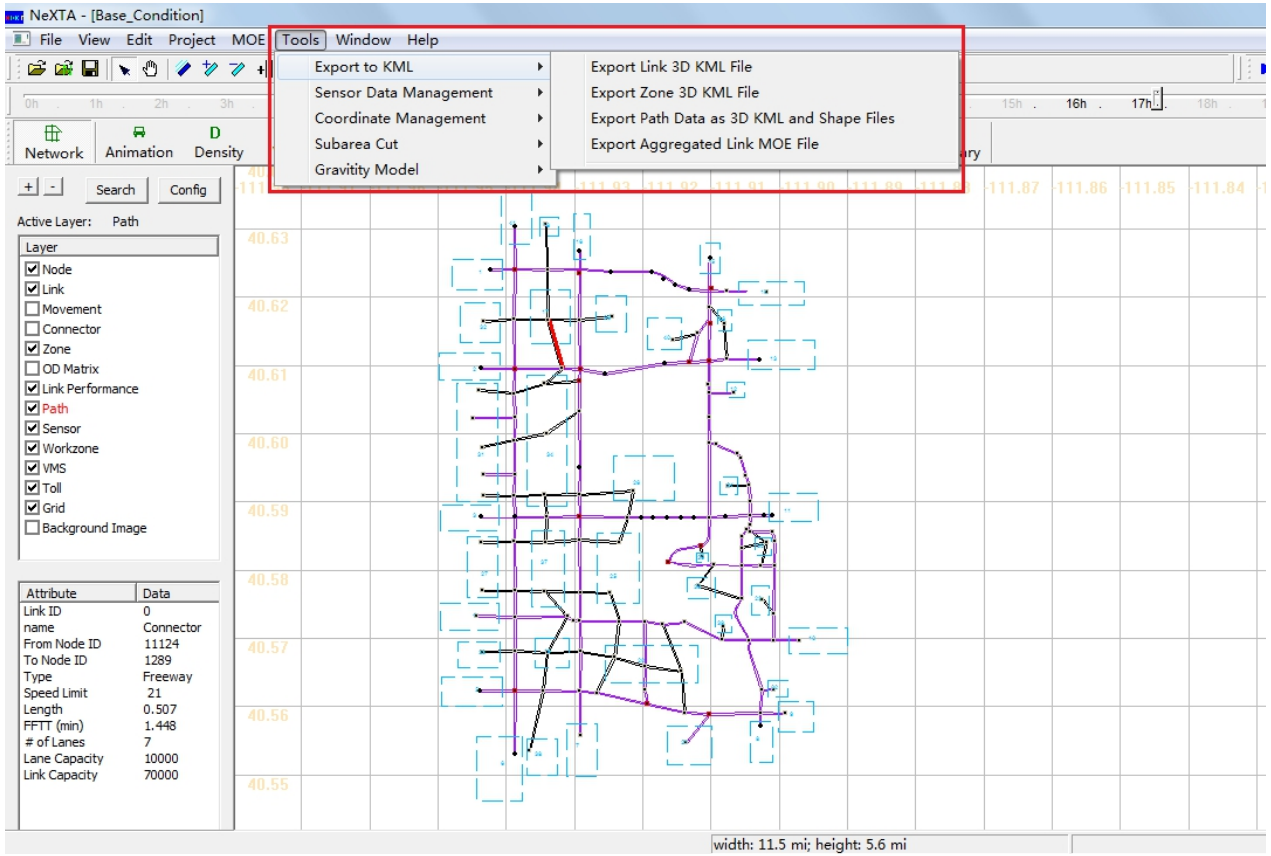
# 3.5 MOE-related display

- Link-related display
- Path-related display
- Vehicle and Summary Charts



# 3.6 Export Data to Google Earth/GIS package

- 2D KML, 3D KML, GIS shape files
- Google Earth visualization
- Zone level display: adjust height/color



# Module 4
# Introduction to Workshop Exercises

4.1 Data Preparation for Building A Regional Traffic Network

4.2 Clackamas Network Modeling(Calibration)

# 4.1 Data Preparation for Building A Regional Traffic Network

| Data Block | File Name |
|---|---|
| **GIS Shape Files** | 1a: node shape file |
| | 1b: link shape file |
| | 1c: zone shape file |
| **GIS Importing Setting Files** | 2: input_GIS_setting.csv |
| **Advanced Definition Files** | 3a: input_node_control_type.csv |
| | 3b: input_link_type.csv |
| | 3c: input_demand_type.csv |
| | 3d: optional_vehicle_emission_rate.csv |
| **Demand Files** | 4.Demand files |
| **Simulation Configuration Files** | 5: input_demand_file_list.csv |
| | 6. input_scenario_settings.csv |
| **D: Scenario Files** | 7a: sensor_count.csv |
| | 7b: sensor_speed.csv |
| | 8: Scenario_Work_Zone.csv |
| | 9: Scenario_Dynamic_Message_Sign.csv |
| | 10: Scenario_Link_Based_Toll.csv |

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 1: Prepare Required Shape files

- node shape file (1a)
- link shape file (1b)
- zone shape file (1c) -- optional

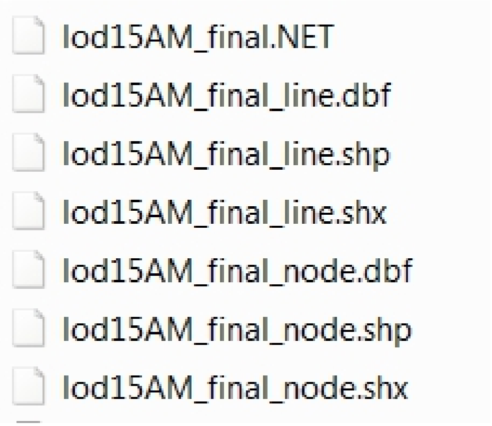
- e.g. given shape files for the Atlanta network:

- lod15AM_final_node.shp:

  provide node ID and coordinate values;

- lod15AM_final_line.shp:

  provide from node id, to node id, length of links, link type, number of lanes of link, ect.



# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 2: Extract GIS information from shape files –importing configuration

- The hub between shape files and NeXTA network data is import_GIS_settings.csv.(2)

- How to do the settings for import_GIS_settings.csv can be found at the learning document in “DTALite-NEXTA-Software-Package”

[3] learning_document_GIS_importing.docx

| key | value |
|---|---|
| node | lod15AM_final_node.shp |
| link | lod15AM_final_line.shp |
| zone | |
| centroid | |
| connector | |
| with_decimal_long_lat | yes |
| length_unit | mile |
| number_of_lanes_oneway_vs_twoway | oneway |
| lane_capacity_vs_link_capacity | lane |
| conversion_factor_for_obtaining_hourly_capacity | 1 |
| direction_0_as_oneway_vs_twoway | twoway |
| default_link_direction | oneway |
| node_number_threshold_as_centroid | 5981 |
| use_default_speed_limit_from_link_type | yes |
| use_default_lane_capacity_from_link_type | yes |
| use_default_number_of_lanes_from_link_type | no |
| identify_from_node_id_and_to_node_id_based_on_geometry | yes |
| create_connectors_for_isolated_nodes | no |
| identify_signal_intersection | no |
| minimum_speed_limit_for_signals | |
| maximum_speed_limit_for_signals | |
| default_cycle_length_in_second | |
| minimum_length_for_importing_links | 0.00001 |
| node_id | N |
| name | |
| TAZ | |
| control_type | |
| from_node_id | A |
| to_node_id | B |

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 3: Meta data files for building the network

- input_link_type.csv (3b)
- input_node_control_type.csv(3a) -- optional

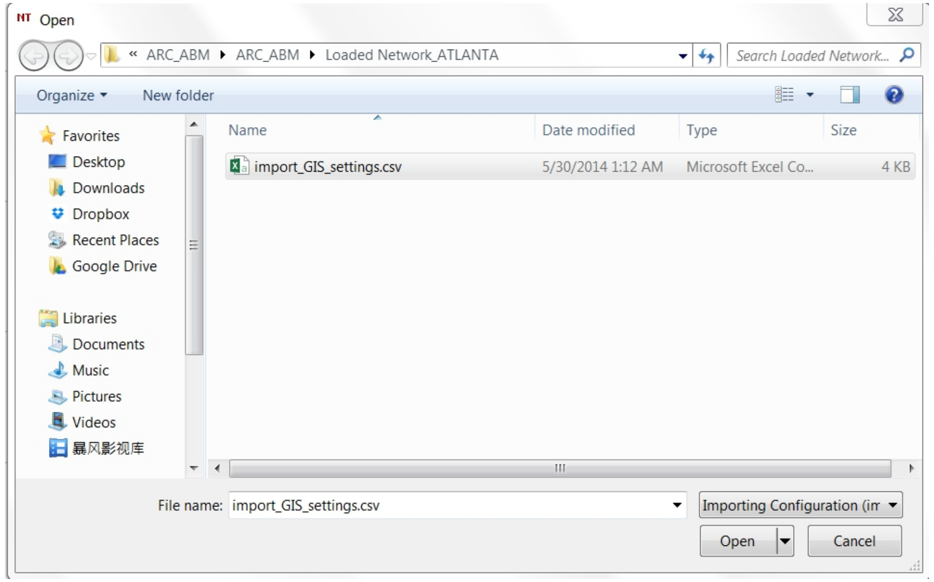
| link_type | link_type_ | type_code | default_la | default_sp | default_nu | capacity_a | travel_tim | approxima |
|---|---|---|---|---|---|---|---|---|
| 0 | centroid c | c | 10000 | 7 | 2 | 1 | 1 | 0 |
| 1 | interstate/ | f | 1900 | 64 | 2 | 1 | 1 | 0 |
| 2 | expresswa | h | 1200 | 43 | 2 | 1 | 1 | 0 |
| 3 | parkway | r | 1150 | 41 | 2 | 1 | 1 | 0 |
| 4 | freeway H | f | 1900 | 66 | 2 | 1 | 1 | 0 |
| 5 | freeway H | f | 1900 | 66 | 2 | 1 | 1 | 0 |
| 6 | freeway tr | f | 1900 | 64 | 2 | 1 | 1 | 0 |
| 7 | system to | r | 1300 | 50 | 2 | 1 | 1 | 0 |
| 8 | exit ramp | r | 800 | 25 | 2 | 1 | 1 | 0 |
| 9 | entrance r | r | 900 | 25 | 2 | 1 | 1 | 0 |
| 10 | principal a | a | 1000 | 29 | 2 | 1 | 1 | 0 |
| 11 | minor arte | a | 900 | 26 | 2 | 1 | 1 | 0 |
| 12 | arterial HC | a | 1000 | 25 | 2 | 1 | 1 | 0 |

| control_type_name | unknown_ | no_contro | yield_sign | 2way_stop | 4way_stop | pretimed_ | actuated_ | roundabout |
|---|---|---|---|---|---|---|---|---|
| control_type | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 100 |

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 4: Import GIS network to NeXTA

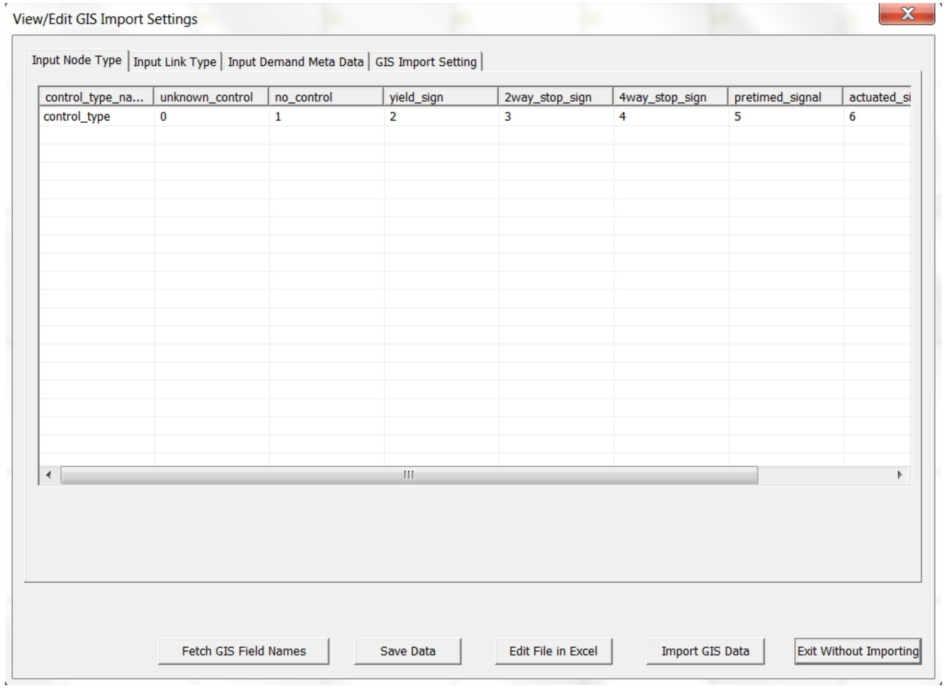
- Open NEXTA_GIS_Import.exe that is the only one program to generate traffic network;
- Click “File” ->“Import GIS Data Set”->“Next”. It will display the following dialogue.



# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 4: Import GIS network to NeXTA

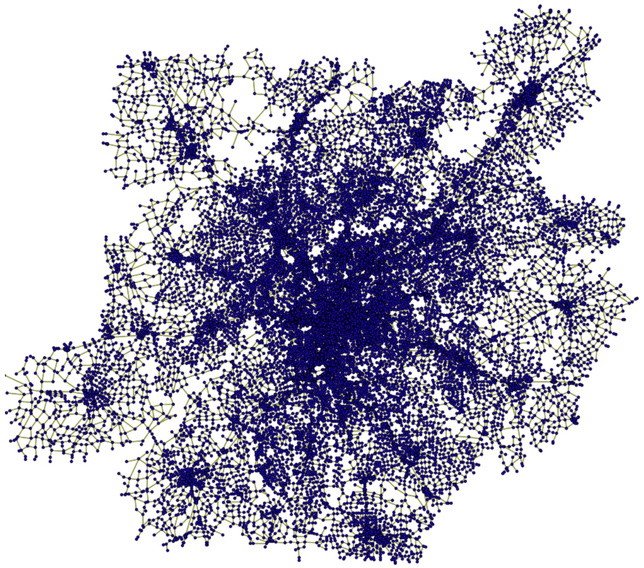
- Choose the import_GIS_setting.csv file and open it. It will display the right interface: those four csv files can be checked here.



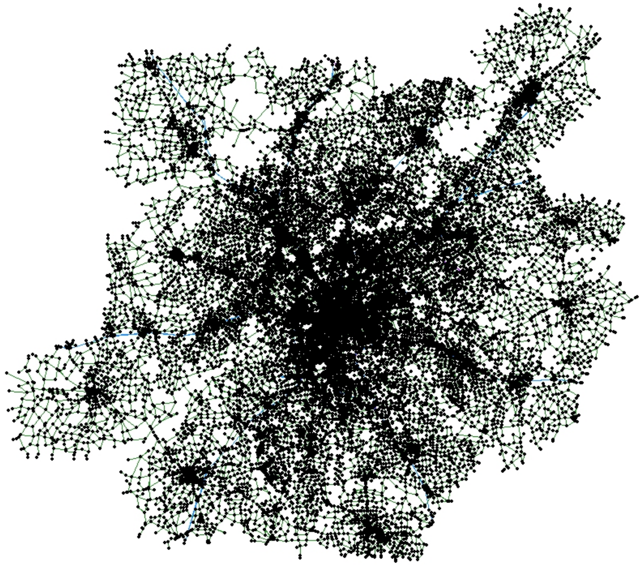
# 4.1 Data Preparation for Building A Regional Traffic Network

### Step 4: Import GIS network to NeXTA

- Click “Import GIS Data” button. And the network will be generated.



GIS Shape files



NeXTA

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 5: Prepare Demand files

- Demand file (4) is the source of dynamic traffic assignment. Generally, it is provided by users. And all types of demand should be defined in Input_demand_type(3c)

- NeXTA can accommodate various demand formats:
- column format from VISUM,
- matrix format,
- full matrix format,
- Dynasmart format,
- agent file,
- and so on.

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 5: Prepare Demand files

- Input_demand_file_list.csv(5) is used to define the characteristics of demand data (4).

| scenario_r | file_seque | file_name | format_ty | number_o | loading_m | start_time | end_time_ | apply_add | subtotal_i | number_o | demand_t | demand_t | demand_t | demand_t |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | htkbypass_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 2 | htkbypass_ | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 3 | htkbypass_ | full_matrix | 0 | 1 | 600 | 900 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 4 | htkbypass_ | full_matrix | 0 | 1 | 900 | 1140 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 5 | mtk_EAMT | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 6 | mtk_AMM | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 7 | mtk_MDM | full_matrix | 0 | 1 | 600 | 900 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 8 | mtk_PMM | full_matrix | 0 | 1 | 900 | 1140 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| 0 | 9 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 10 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 11 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 12 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 13 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 14 | TODEA15_ | full_matrix | 0 | 1 | 180 | 360 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 15 | TODAM15 | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 16 | TODAM15 | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 17 | TODAM15 | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 18 | TODAM15 | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| 0 | 19 | TODAM15 | full_matrix | 0 | 1 | 360 | 600 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 6: Data required for OD Demand calibration

- sensor_count.csv (7a)
- link_count , matched_link_from_node_id, matched_link_to_node_id;

| count_sen | day_no | start_time | end_time_ | direction | link_count | derived_la | matched_l | matched_link_from_node_id | matched_link_to_node_id | matched_l |
|---|---|---|---|---|---|---|---|---|---|---|
| 25008_2 | 16 | 375 | 390 | 2 | 186 | 248 | 626 | 553 | Major arterial | 3 |
| 24002_2 | 16 | 435 | 450 | 2 | 589 | 785.3333 | 594 | 599 | Major arterial | 3 |
| 41009_2 | 16 | 450 | 465 | 2 | 526 | 526 | 691 | 694 | Major arterial | 4 |
| 3051_2 | 16 | 480 | 495 | 2 | 771 | 1028 | 2310 | 2253 | RR 3&4 | 3 |
| 24003_2 | 16 | 360 | 375 | 2 | 252 | 336 | 590 | 593 | Major arterial | 3 |
| 41001_2 | 16 | 405 | 420 | 2 | 935 | 1246.667 | 1860 | 1897 | Urban Express | 3 |
| 4032_2 | 16 | 495 | 510 | 2 | 1579 | 1579 | 357 | 314 | RR 3&4 | 4 |
| 41005_2 | 16 | 375 | 390 | 2 | 325 | 433.3333 | 1936 | 1901 | Urban Express | 3 |
| 4032_2 | 16 | 405 | 420 | 2 | 1238 | 1238 | 357 | 314 | RR 3&4 | 4 |
| 3046_2 | 16 | 435 | 450 | 2 | 1631 | 2174.667 | 1888 | 1812 | RR 3&4 | 3 |
| 3046_2 | 16 | 405 | 420 | 2 | 1665 | 2220 | 1888 | 1812 | RR 3&4 | 3 |
| 4039_2 | 16 | 510 | 525 | 2 | 1151 | 1151 | 660 | 640 | RR 3&4 | 4 |
| 3046_2 | 16 | 525 | 540 | 2 | 1632 | 2176 | 1888 | 1812 | RR 3&4 | 3 |
| 25006_2 | 16 | 420 | 435 | 2 | 638 | 850.6667 | 2521 | 2484 | Major arterial | 3 |
| 2050_2 | 16 | 360 | 375 | 2 | 1032 | 1032 | 1653 | 1616 | RR 2nd | 4 |
| 3046_2 | 16 | 495 | 510 | 2 | 1611 | 2148 | 1888 | 1812 | RR 3&4 | 3 |
| 3051_1 | 16 | 390 | 405 | 1 | 625 | 833.3333 | 2243 | 2304 | RR 3&4 | 3 |
| 3055_1 | 16 | 495 | 510 | 1 | 1068 | 1424 | 2353 | 2379 | RR 3&4 | 3 |

Sensor_count.csv

- sensor_speed.csv(7b)
- speed_sensor_id , speed, population_count;

| day_no | start_time | end_time_ | speed_sen | speed | probe_cou | population_count |
|---|---|---|---|---|---|---|
| 16 | 360 | 375 | 14988 | 33.3459 | 4 | 40 |
| 16 | 360 | 375 | 32655 | 46.33333 | 1 | 10 |
| 16 | 360 | 375 | 7185 | 56.52771 | 2 | 20 |
| 16 | 360 | 375 | 125928 | 31.29616 | 2 | 20 |
| 16 | 360 | 375 | 24497 | 71.30745 | 1 | 10 |
| 16 | 360 | 375 | 22459 | 73.44187 | 1 | 10 |
| 16 | 360 | 375 | 39107 | 51.83703 | 1 | 10 |
| 16 | 360 | 375 | 28241 | 64.96103 | 2 | 20 |
| 16 | 360 | 375 | 17696 | 49.60911 | 7 | 70 |
| 16 | 360 | 375 | 46810 | 42.01696 | 5 | 50 |

sensor_speed.csv

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 7: Work zone evaluation

- The setting file for work zone simulation is
- Scenario_Work_Zone.csv (8)

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| Link | Scenario No | Start Day | End Day | Start Time in Min | End Time in min | Capacity Reduction Percentage (%) | Speed Limit (mph) |
| 1 | 1 | 1 | 30 | 420 | 540 | 20 | 40 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Scenario_Work_Zone.csv

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 8: VMS evaluation

- Scenario_Dynamic_Message_Sign.csv(9) is used to define the location and characteristics of variable message signs in the simulation

| A | B | C | D | E | F | G | H | |
|---|---|---|---|---|---|---|---|---|
| Link | Scenar | Start Day | End Day N | Start Tim | End Time i | Number oi | Detour Route 1 | De |
| [270387, 242201] | 0 | 10 | 10 | 450 | 510 | 1 | 10;242201;242202;233800;244347;244346;236589;270393;266159;266160;283899;0.80 | |

Scenario_Message_Sign.csv

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 9: Tolling evaluation

- Scenario_Link_Based_Toll.csv (10)
- Since the toll for different demand types is represented by money, the data Value of Time in input_demand_type.csv is important.

| Link | Scenario No | Start Day | End Day | Start Time in Min | End Time in min | Charge for LOV ($) | Charge for HOV ($) | Charge for Truck ($) | Charge for Intermodal ($) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 30 | 420 | 540 | 0.5 | 0.2 | 0.2 | 0.3 |
| 2 | 1 | 1 | 30 | 420 | 540 | 0.5 | 0.2 | 0.2 | 0.3 |
| 3 | 1 | 1 | 30 | 420 | 540 | 0.5 | 0.2 | 0.2 | 0.3 |
| 4 | 1 | 1 | 30 | 420 | 540 | 0.5 | 0.2 | 0.2 | 0.3 |
| 5 | 1 | 1 | 30 | 420 | 540 | 0.5 | 0.2 | 0.2 | 0.3 |

Scenario_Link_Based_Toll.csv

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 10: Emission evaluation(9a)

- optional_vehicle_emission_rate.csv (3d)

| vehicle_ty | OpModelD | meanBase | meanBase | meanBase | meanBase | meanBase | age |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 49206.3 | 3536.293 | 0.05385 | 2.36609 | 0.039171 | 0 |
| 1 | 1 | 45521.4 | 3271.471 | 0.008979 | 4.05557 | 0.000418 | 0 |
| 1 | 11 | 71581.4 | 5144.317 | 0.146868 | 6.52187 | 0.022892 | 0 |
| 1 | 12 | 98841 | 7103.373 | 0.155233 | 2.82379 | 0.02085 | 0 |
| 1 | 13 | 137367 | 9872.108 | 0.363034 | 9.76815 | 0.052262 | 0 |
| 1 | 14 | 173571 | 12473.97 | 0.657844 | 14.2137 | 0.072532 | 0 |
| 1 | 15 | 206979 | 14874.89 | 1.18797 | 20.8813 | 0.103686 | 0 |
| 1 | 16 | 249989 | 17965.88 | 2.5348 | 35.9857 | 0.171948 | 0 |
| 1 | 21 | 97382.5 | 6998.556 | 0.254133 | 5.8165 | 0.038721 | 0 |

optional_vehicle_emission_rate.csv

# 4.1 Data Preparation for Building A Regional Traffic Network

## Step 11: Run Simulation

- Input_scenario_setting.csv (6)

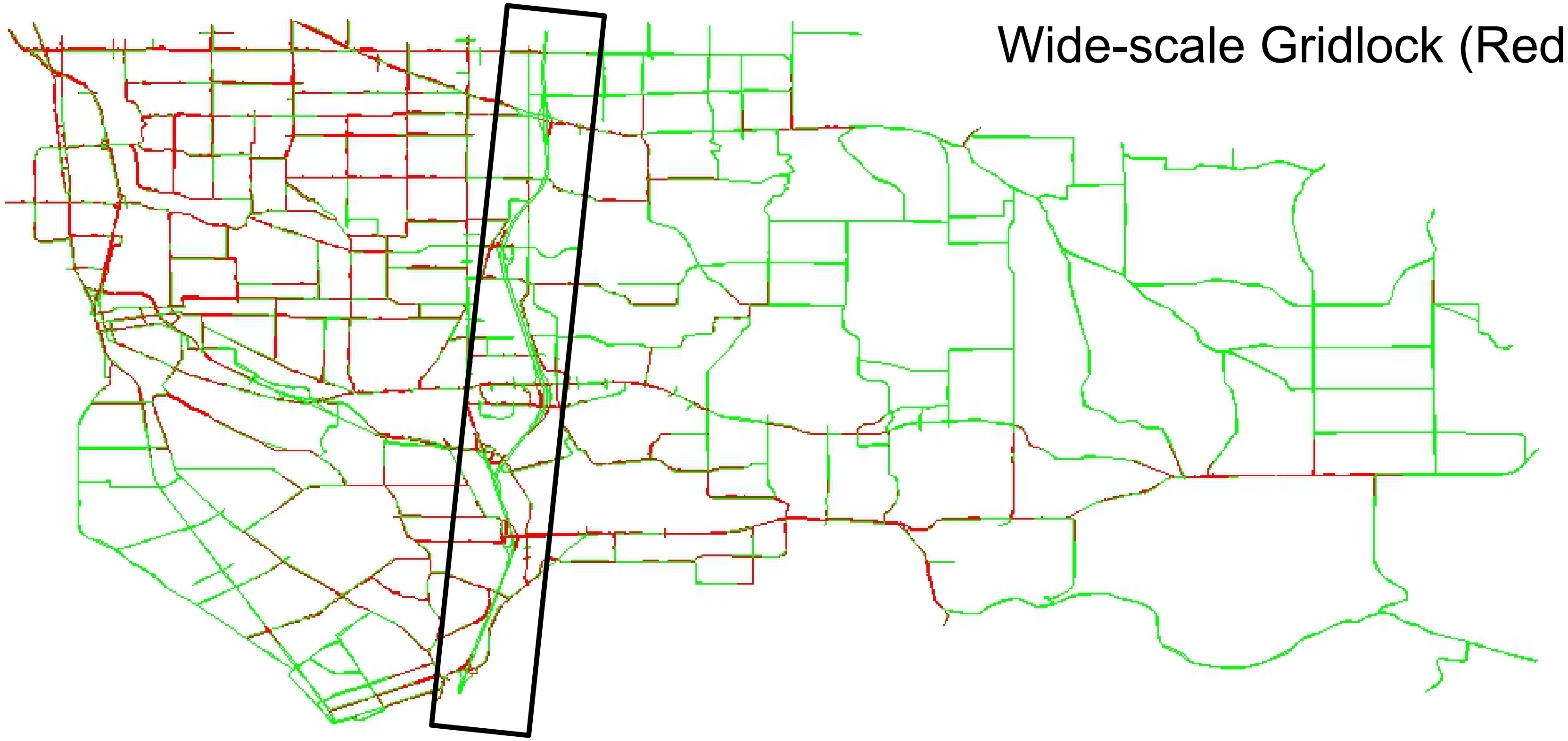
| A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|
| scenario_ | scenario_na | number_of_ite | traffic_f | signal_repr | traffic_ana | random_seed | ODME_star | ODME_max_ | ODME_step | calibrati | calibratio |
| 1 | test1 | 3 | 1 | 0 | 0 | 100 | 20 | 50 | 0.15 | 990 | 1050 |

Input_scenario_setting.csv

# 4.2 Clackamas Network Modeling(Calibration)

## Converting from Macroscopioc Model to Mesoscopic Model

- Directly import important network attributes: Capacity, speed, number of lanes, etc.
- First Simulation Run:

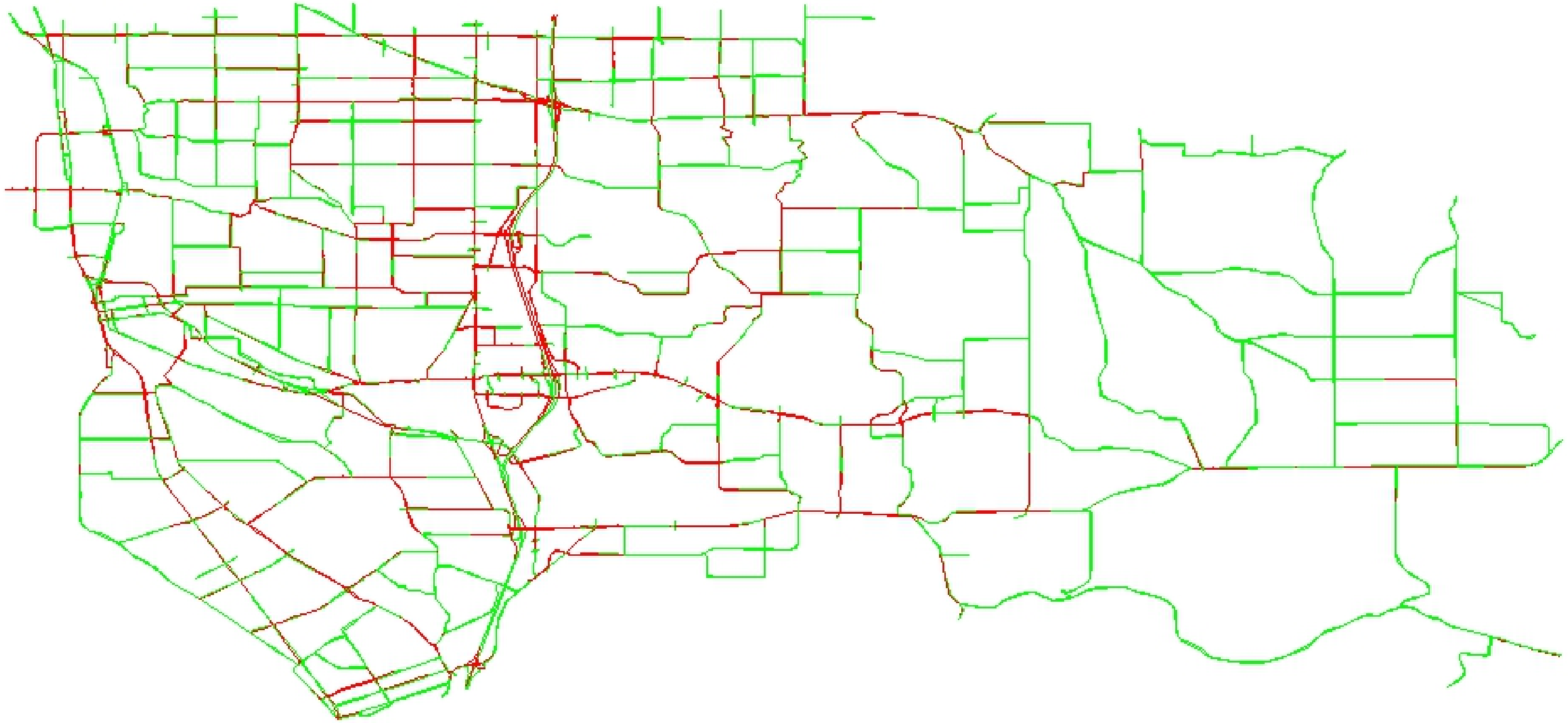


Wide-scale Gridlock (Red)

# 4.2 Clackamas Network Modeling(Calibration)

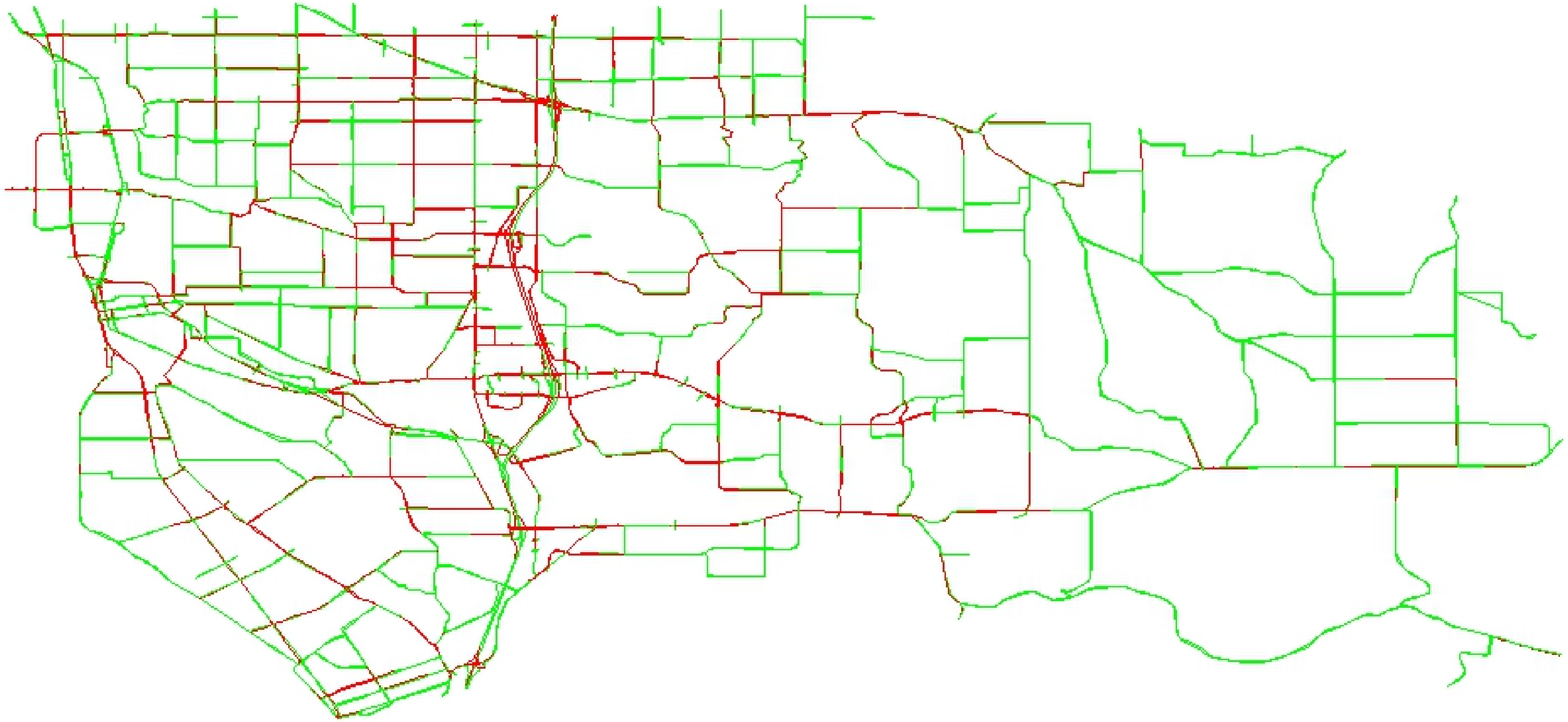
**Second Attempt**

- Increased ramp outflow capacity
  - Still experiencing significant queuing



# 4.2 Clackamas Network Modeling(Calibration)

**Inflow/Storage Capacity?**



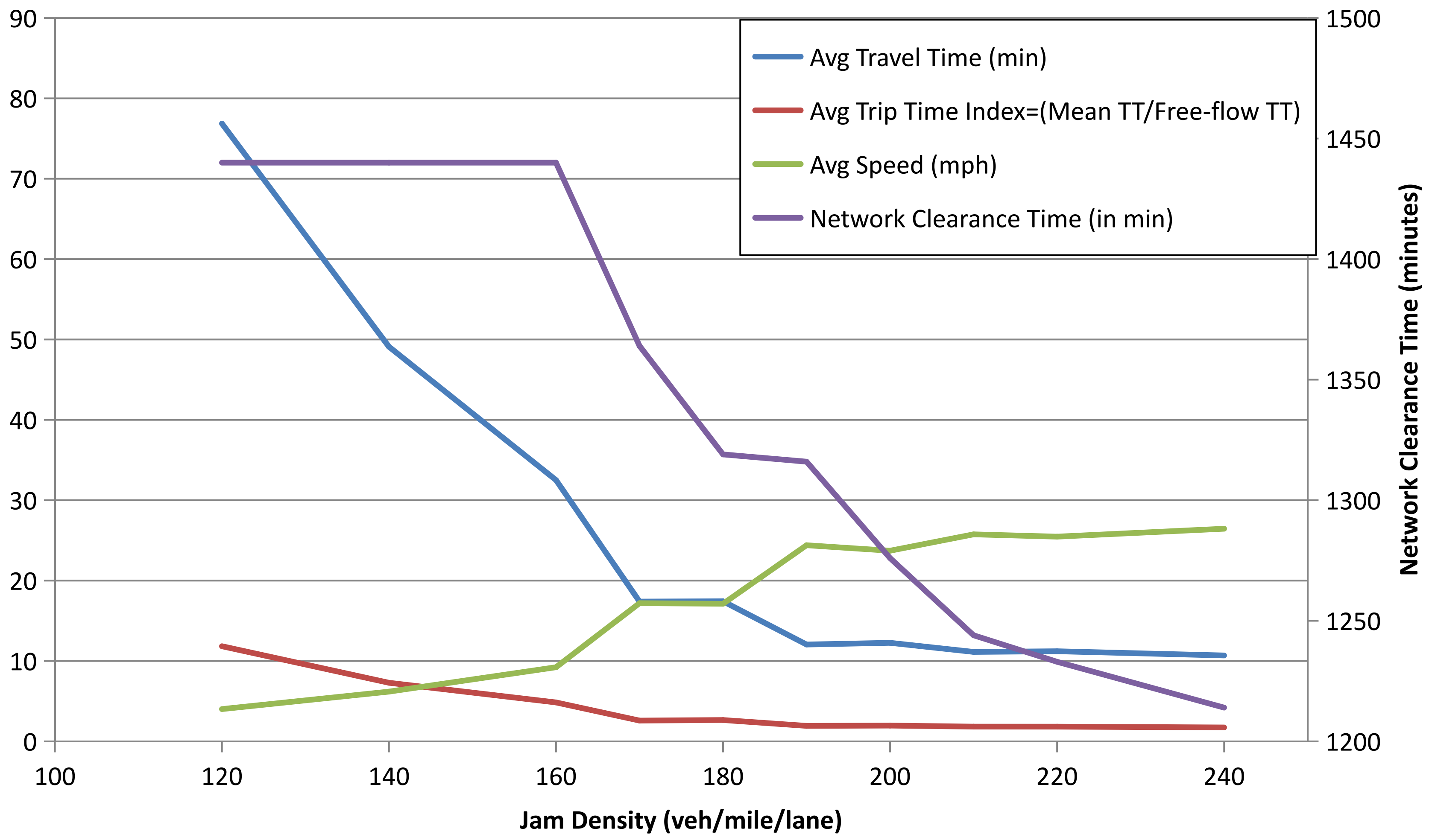
# 4.2 Clackamas Network Modeling(Calibration)

## Geometry Details

- Two-lane ramp, coded with one lane
  - Reasonable outflow capacity
  - Potential issues
- Underestimated inflow capacity
  - Underestimated storage capacity

# 4.2 Clackamas Network Modeling(Calibration)

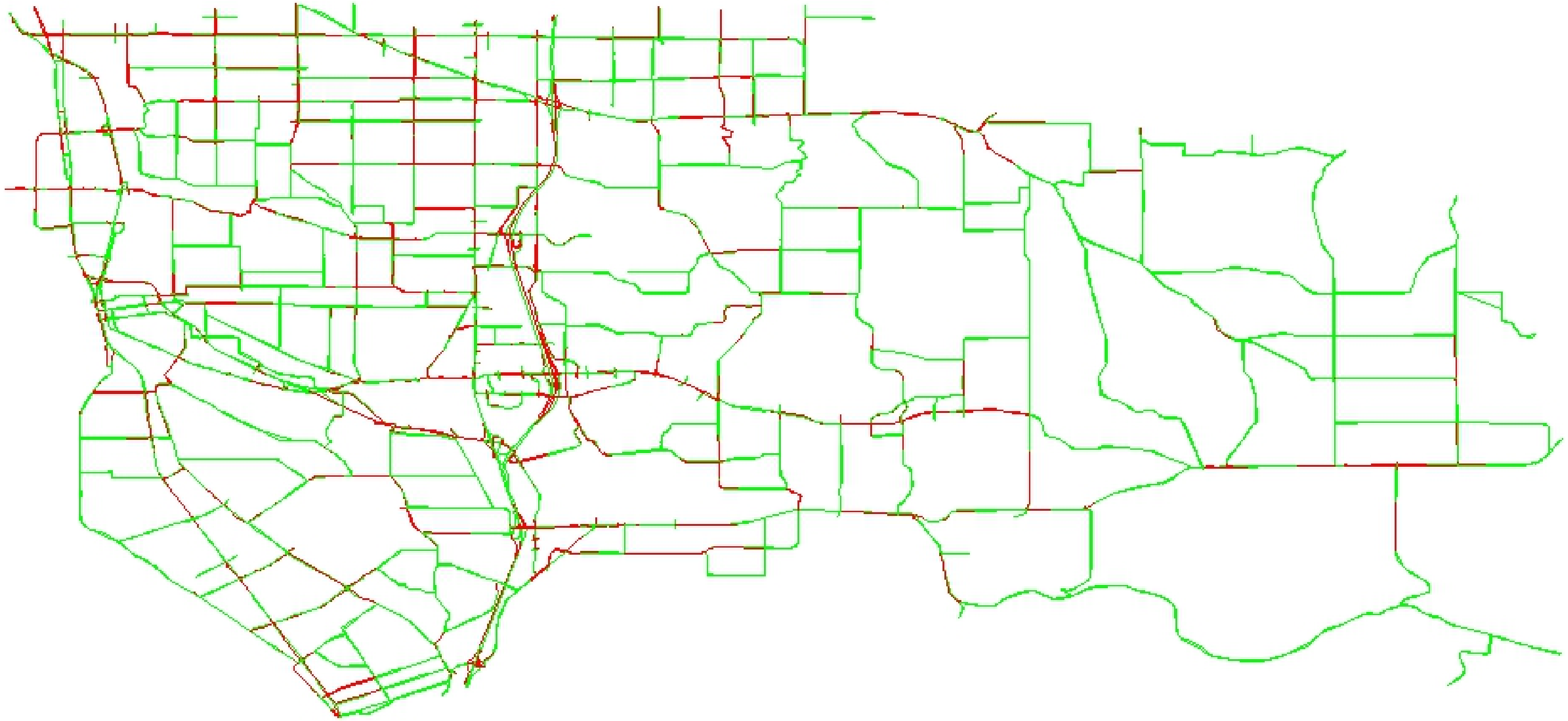
## Traffic Flow Model Sensitivity



# 4.2 Clackamas Network Modeling(Calibration)

**Third Attempt**

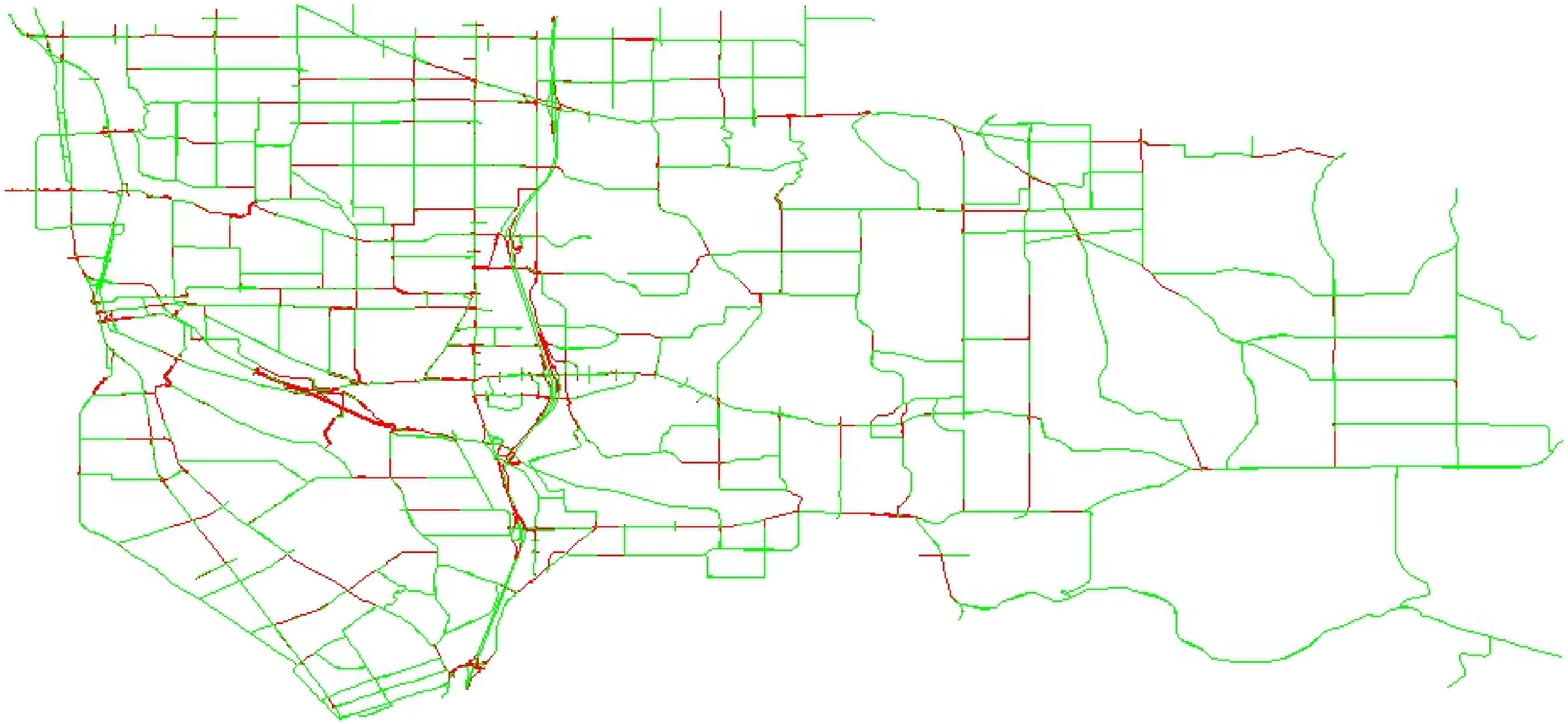
- Reset outflow capacity, adjusted inflow & storage capacity



# 4.2 Clackamas Network Modeling(Calibration)

## Combined Modifications

- Combination of adjusting outflow and storage capacity appears more reasonable



# 4.2 Clackamas Network Modeling(Calibration)

**Recommended Diagnostic Procedures**

- Macroscopic capacity may not be appropriate for mesoscopic capacity constraints
- Understand the traffic flow model
  - Understand limitations, special cases
- Adjust capacity before OD demand, path flow
- Start with fewer capacity constraints to remove possible unrealistic bottlenecks
  - Point queue → Spatial queue → Shock wave → Speed-density relationships

# Thank You!

**Questions, Comments, and Suggestions are Welcome.**

## Please Contact:

**Xuesong Zhou, Ph.D.**
**Arizona State University**
**Tempe, AZ 85287**
**Email: xzhou74@asu.edu**

# Open-Source Dynamic Traffic Assignment Package

# DTALite/NEXTA

# (September, 2017)

**Xuesong Zhou**

xzhou99@gmail.com

xzhou74@asu.edu

# Outline

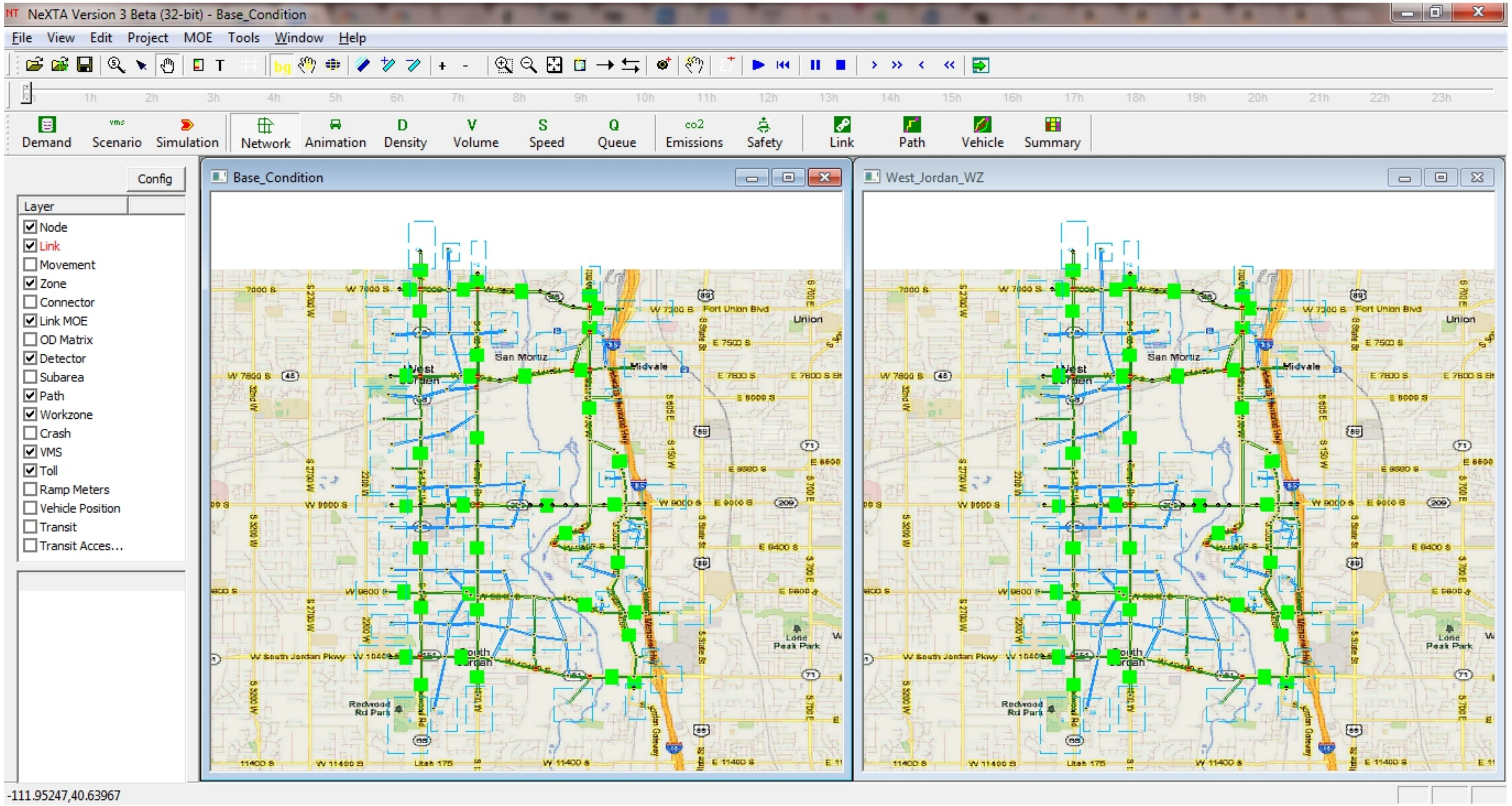
Module 5: Traffic Flow Theory

Module 6: OD Matrix Estimation

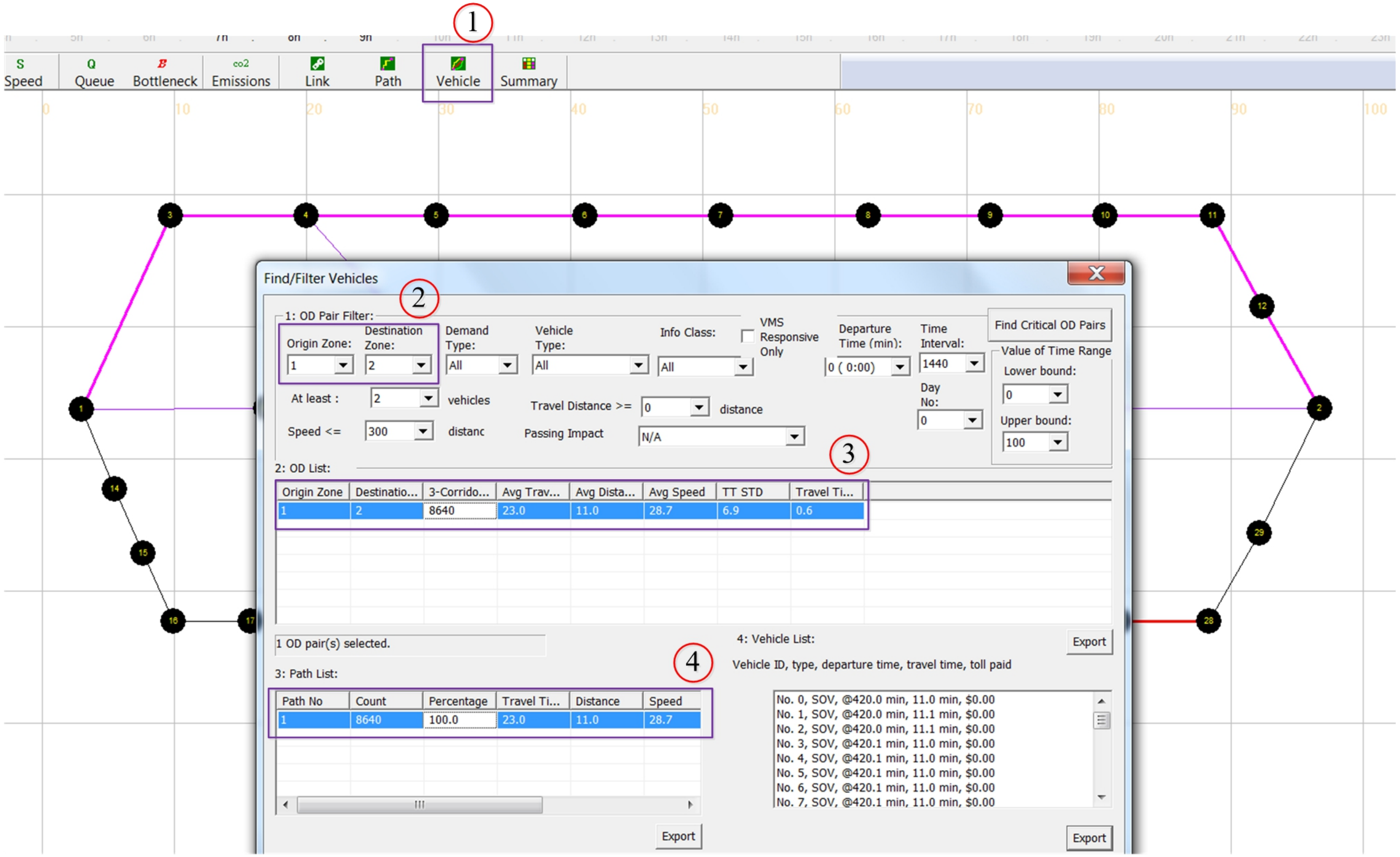
Module 7: Signalized Intersections Modeling

Module 8: Vehicle Routing C++ Code

- Synchronized display in NeXTA



# Vehicle Analysis Dialog



# Module 3: Modeling Approach

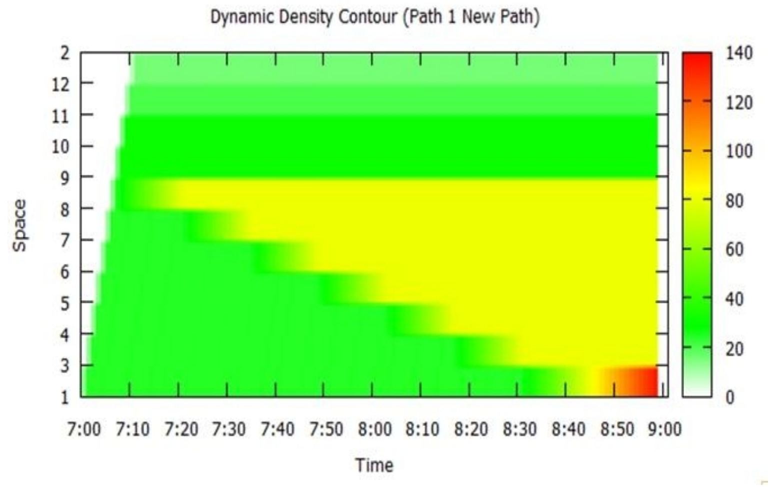
# Module 5
# Traffic Flow Theory

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

Contributing Author: Jiangtao Liu

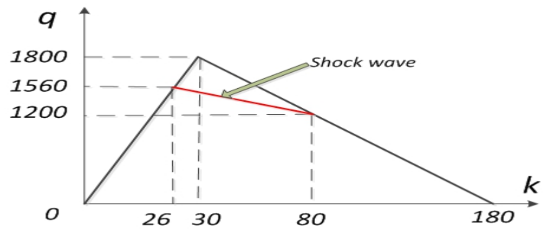
https://scholar.google.com/citations?user=VLP4r4MAAAAJ&hl=en

# 5.1 Traffic Congestion Propagation




## Learning Objectives

- (1) Understand major input and output data for a dynamic network loading program;
- (2) Identify bottlenecks and model congestion propagation;
- (3) Calculate traffic states through different computational approaches.

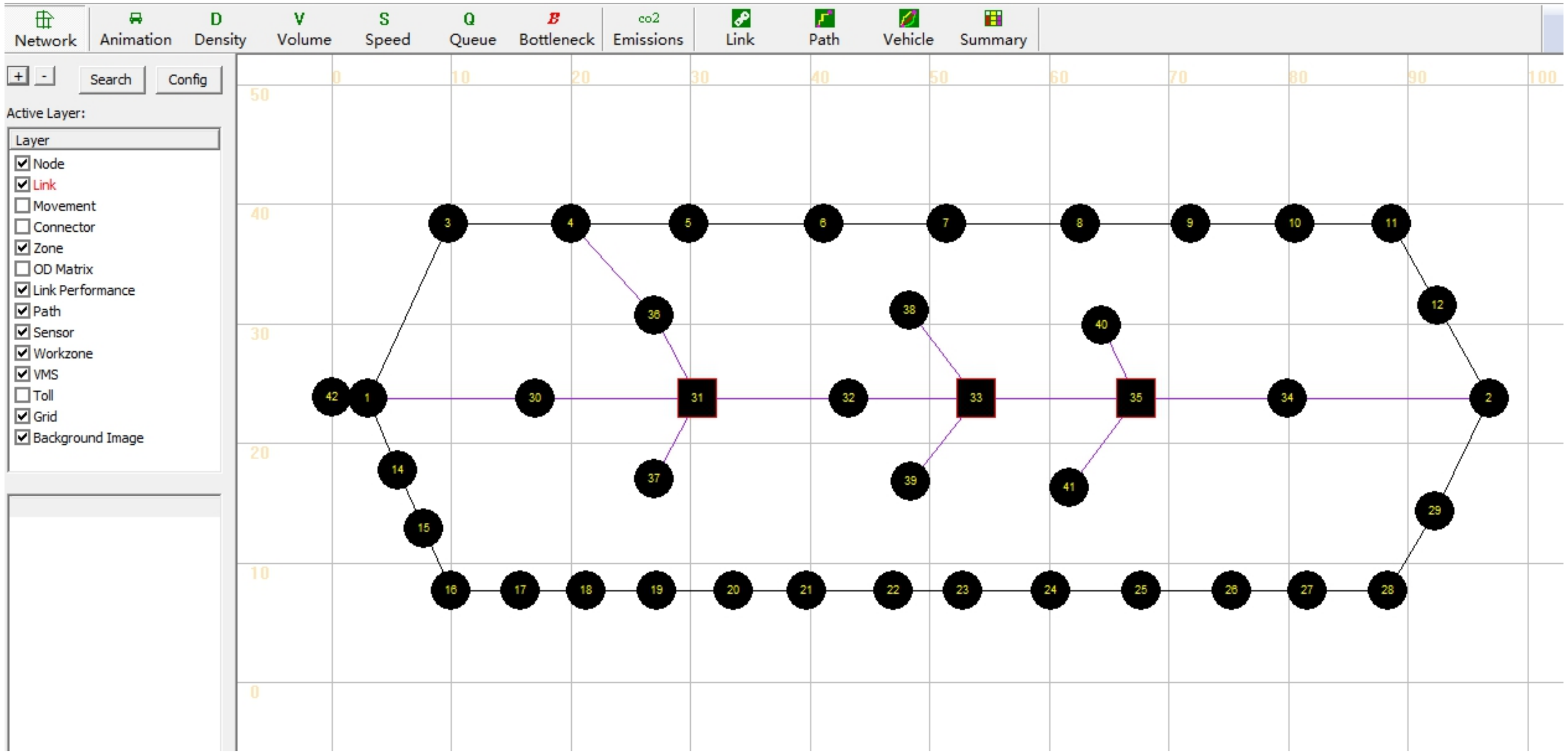



## Data Set

https://github.com/xzhou99/learning-transportation/tree/master/Lessons/Lesson%202/Data%20Set

# 5.1 Traffic Congestion Propagation

## Sample Case: 3-corridor Network



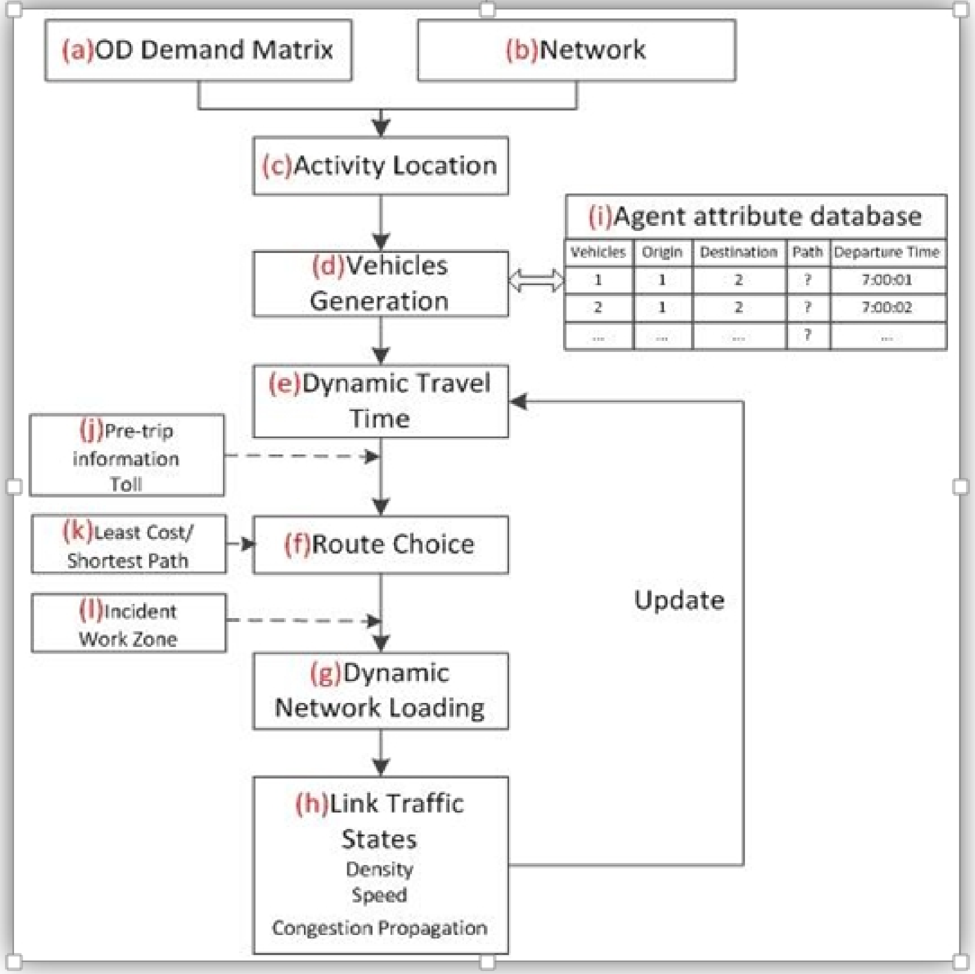



Fig.1. The process of dynamic network loading

1. The **basic data input** for a dynamic network loading program (g) includes time-dependent origin-destination demand (a) and the traffic network (b) with road capacity constraints on links and nodes.

2. **Activity locations** (c) describe a zone-to-node mapping between OD demand matrix (a) and the traffic network (b). The vehicles generation process (d) determines specific origin, destination, and departure time of agents/vehicles, which leads to an agent attribute database (i), to be iteratively updated throughout the simulation process.

3. Through calculating the **dynamic travel time** of all links (free-flow travel time is used at first iteration), corresponding to element (e), the **route choice model** (f) embeds a standard time-dependent least-cost path algorithm (k) to generate paths for all agents.

4. The core **dynamic network loading program** (g) loads previously generated agents on the traffic network for the entire planning horizon, which produces traffic data output (h) that describes time-varying traffic states at the link level, including traffic flow volume, traffic density, speed, travel time. The link-based traffic states (h) can be further processed to represent traffic congestion and propagation at the aggregated path level, as well as update the time-dependent travel time database (e).

Table 1. Properties of traffic network

| File No. | GIS Layer | Associated Data File | Associated Menu for Data Editing | Important Attributes |
|---|---|---|---|---|
| 1 | Node | input_node.csv | Project->Network Data-> Node | node coordinate, control type |
| 2 | Link | input_link.csv | Project->Network Data-> Link | from node, to node, speed limit -> free-flow travel time, capacity, number of lanes |
| 3 | Zone | input_zone.csv | Project->Network Data-> Zone | zone definition for OD demand |
| 4 | Activity Location (similar to centroid) | input_activity_location.csv | Project->Network Data-> Activity Location | mapping from zone number to nodes as origin or destination |
| 5 | OD demand matrix | input_demand_file_list.csv input_demand.csv | Project-> Demand Database | # of trips from zone i to zone j |
| 6 | Traffic simulation setup | input_scenario_settings.csv | Project-> Perform Traffic Assignment | # of iterations, traffic models, traffic assignment method… |
| 7 | Generated Vehicle Data | output_agent.csv | --- | Agent ID, Arrival time, departure time… |
| 8 | Traffic States | output_LinkTDMOE.csv | --- | Travel time, speed, density… |

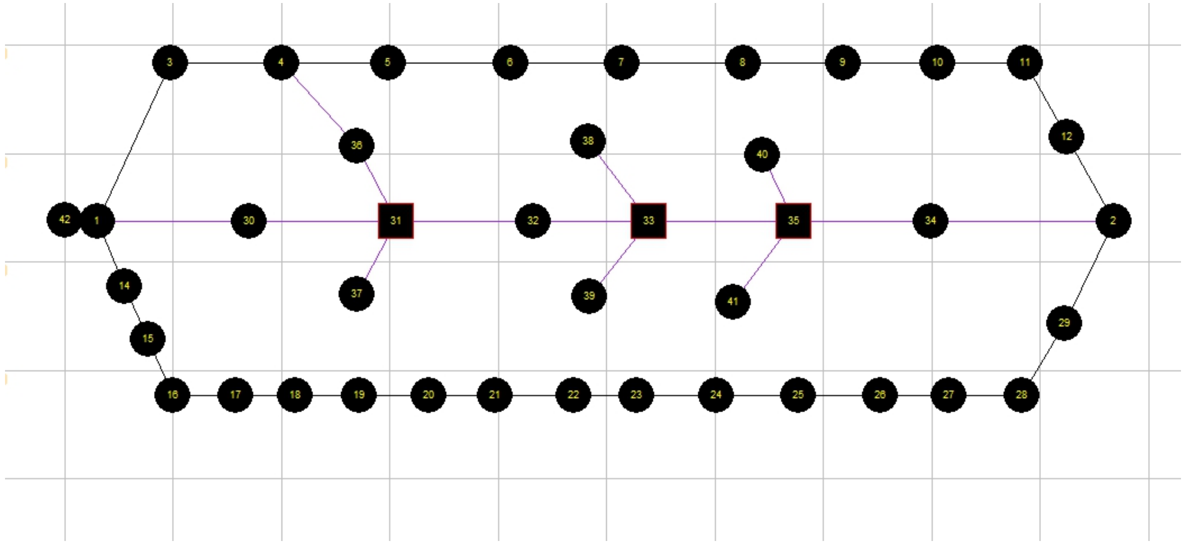



Table 2. Link attributes for calculating the travel time of path 1

| No. | Link (From node-> to Node) | Distance (mile) | Free-flow travel time(min) | Link capacity (veh/h) |
|---|---|---|---|---|
| 1 | 1->3 | 1 | 1 | 5700 |
| 2 | 3->4 | 1 | 1 | 5400 |
| 3 | 4->5 | 1 | 1 | 5400 |
| 4 | 5->6 | 1 | 1 | 5400 |
| 5 | 6->7 | 1 | 1 | 5400 |
| 6 | 7->8 | 1 | 1 | 5400 |

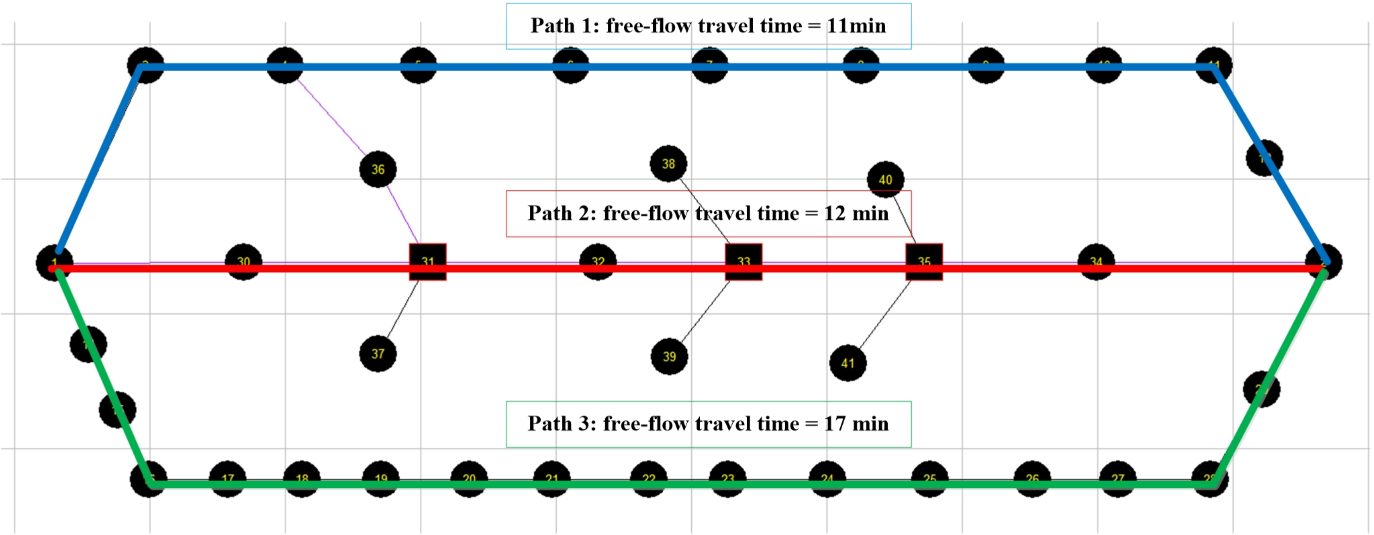



Table 3. Travel time and travel distance of three paths

| Path | Node sequence | # of links | Travel time (min) | Travel Distance (mile) | Bottleneck |
|---|---|---|---|---|---|
| Path 1 (Major Freeway) | 1->3->4...-> 12->2 | 11 | 11 | 11 | Link 9->10, Link 10->11 |
| Path 2 (Alternative Arterial corridor) | 1->30->31...-> 34->2 | 7 | 12 | 8 | |
| Path 3 (Alternative Freeway) | 1->14->15...-> 29->2 | 17 | 17 | 17 | |

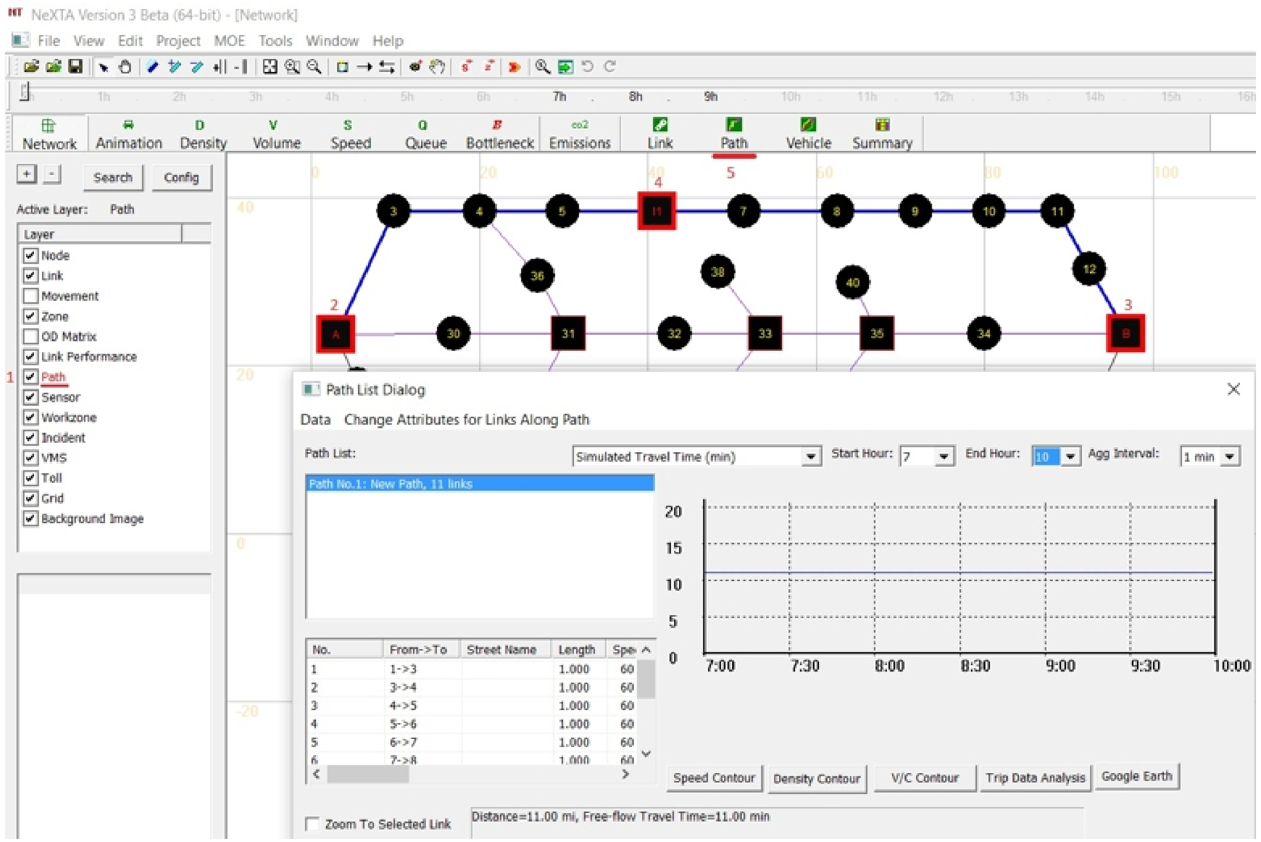



Fig.6. Path list dialog

STEPS

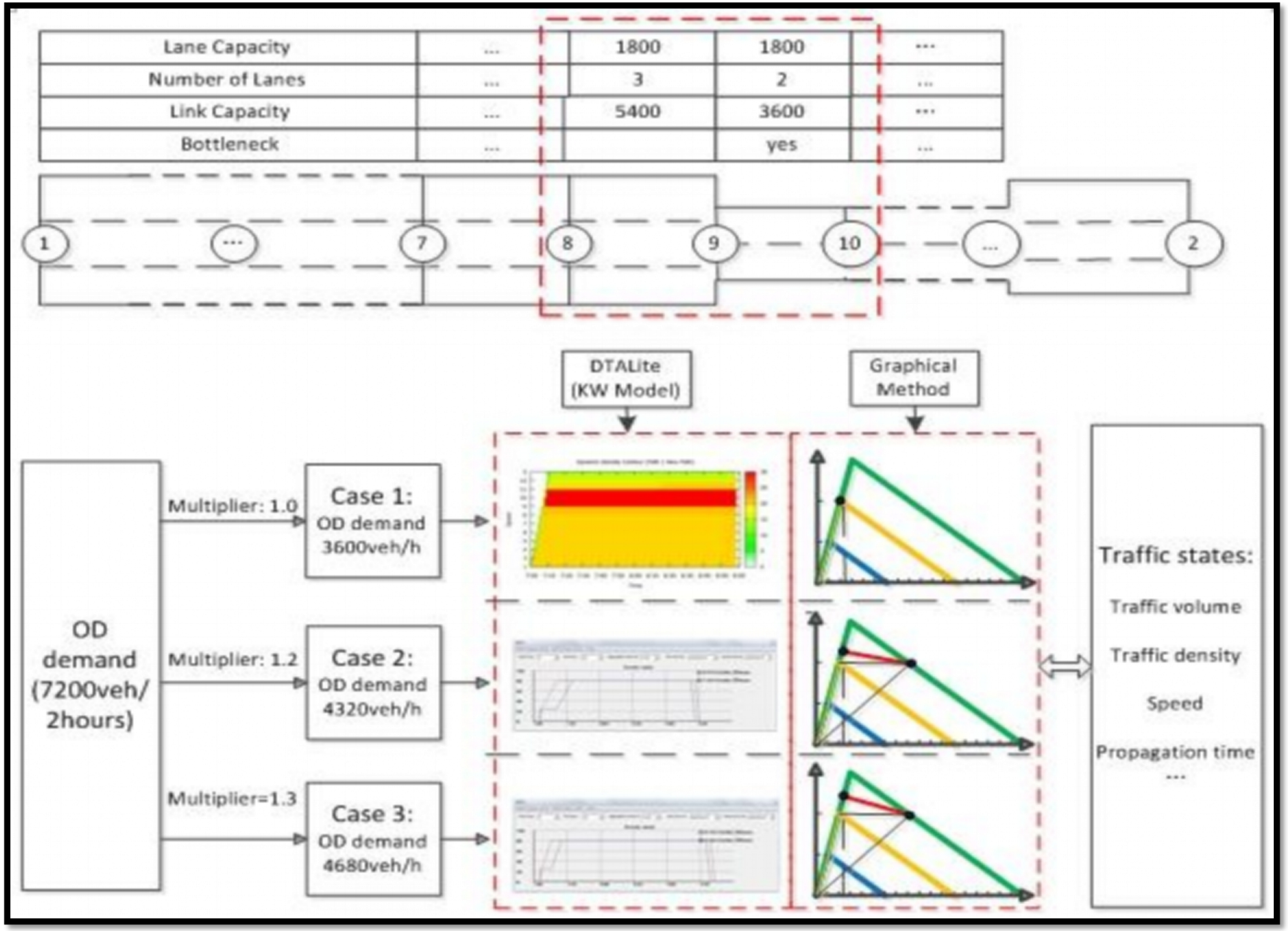
**Step 1**: Select the "path" layer in the GIS layers panel

**Step 2**: Click on node 1-> right click menu to choose "Direction from here"

**Step 3**: Click node 2-> right click menu to choose "Direction to here"

**Step 4**: Select any node (e.g. node 6) of the path 1-> right click to choose "Add Intermediate Destination here".

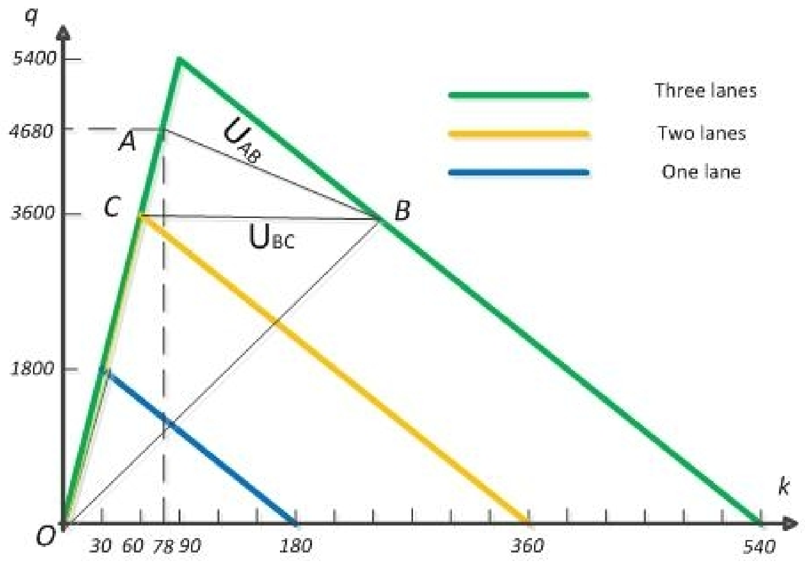
**Step 5**: To view the traffic states of selected path, click on the "Path" button in the MOE bar


Lane Capacity
Number of Lanes
Link Capacity
Bottleneck
1800
1800
3
2
5400
3600
yes
1
7
8
9
10
2
DTALite
(KW Model)
Graphical
Method
OD
demand
(7200veh/
2hours)
Multiplier: 1.0
Multiplier: 1.2
Multiplier=1.3
Case 1:
OD demand
3600veh/h
Case 2:
OD demand
4320veh/h
Case 3:
OD demand
4680veh/h
Traffic states:
Traffic volume
Traffic density
Speed
Propagation time

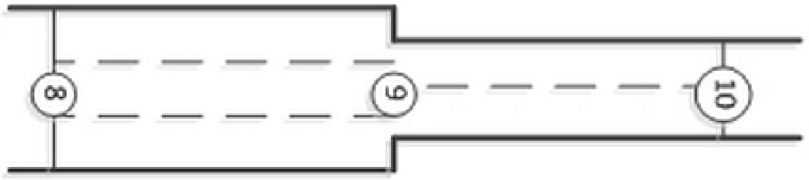
# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

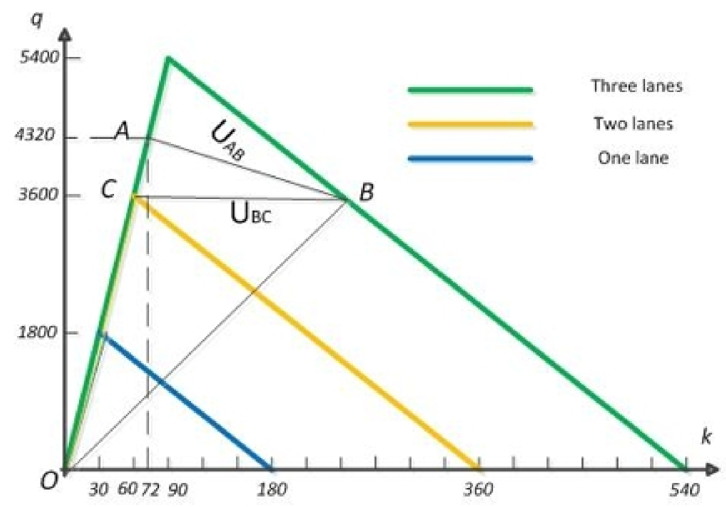
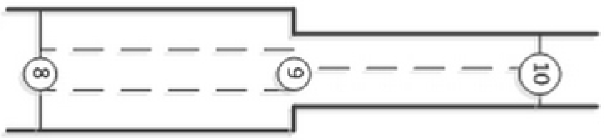
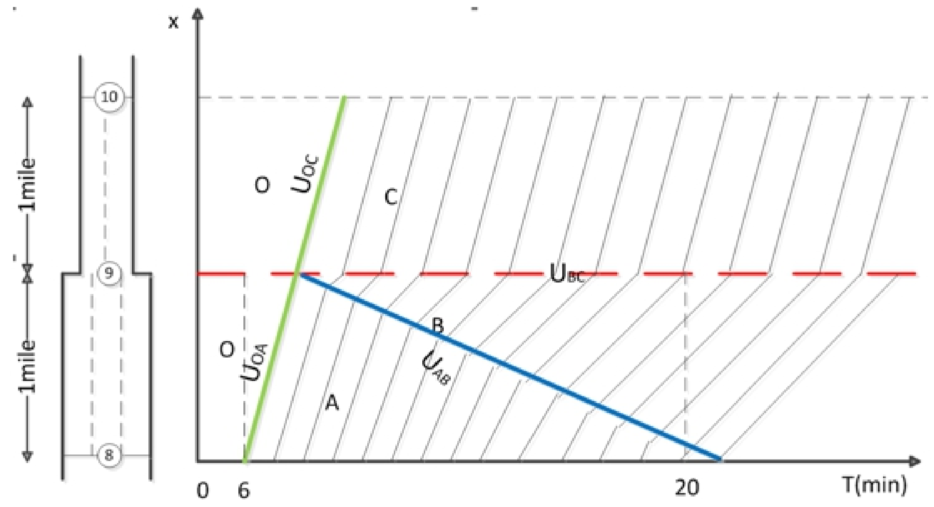
### Graphical Method




The flow-density relationship for different states of links




- U: interface between two different traffic states.
- State O: there is no traffic flow
- State A: the demand traffic enter the link 8->9
- State C: the steady traffic state of link 9->10

| | |
|---|---|
| <br>Fig.29. The flow-density relationship for different states of links | <br>U: interface between two different traffic states.<br><br>State O: there is no traffic flow<br><br>State A: the demand traffic enters the link 8->9<br><br>State B: the steady traffic state of link 8->9<br><br>State C: the steady traffic state of link 9->10 |
| <br>Fig.30. Vehicles' trajectory in time-space diagram | **Speed:** for link 8->9, at state A, its speed is 60 mph; at state B, due to the limitation of capacity, its speed is 15 mph. For link 9->10, the vehicles will drive at a speed based on the capacity, state C, whose speed is 60 mph;<br><br>**Density:** for link 8->9, at the state A, its density is 72 vpmpl(link). After the propagation time, it is at the state B, whose density is 240 vpmpl(link). Meanwhile, the link 9->10 is always at the state C, whose density is 60 vpmpl(link). In addition, the method of calculating the density of each lane is the same as case 1.<br><br>**Propagation time:** 14 min. |

# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 1: Check the value of traffic demand

- When Demand multiplier is 1.0, the hourly OD demand is 0.5*7200*1 = 3600 vehicles.
- The limit capacity of the path 1 is 3600, so now the demand is equal to the path capacity.

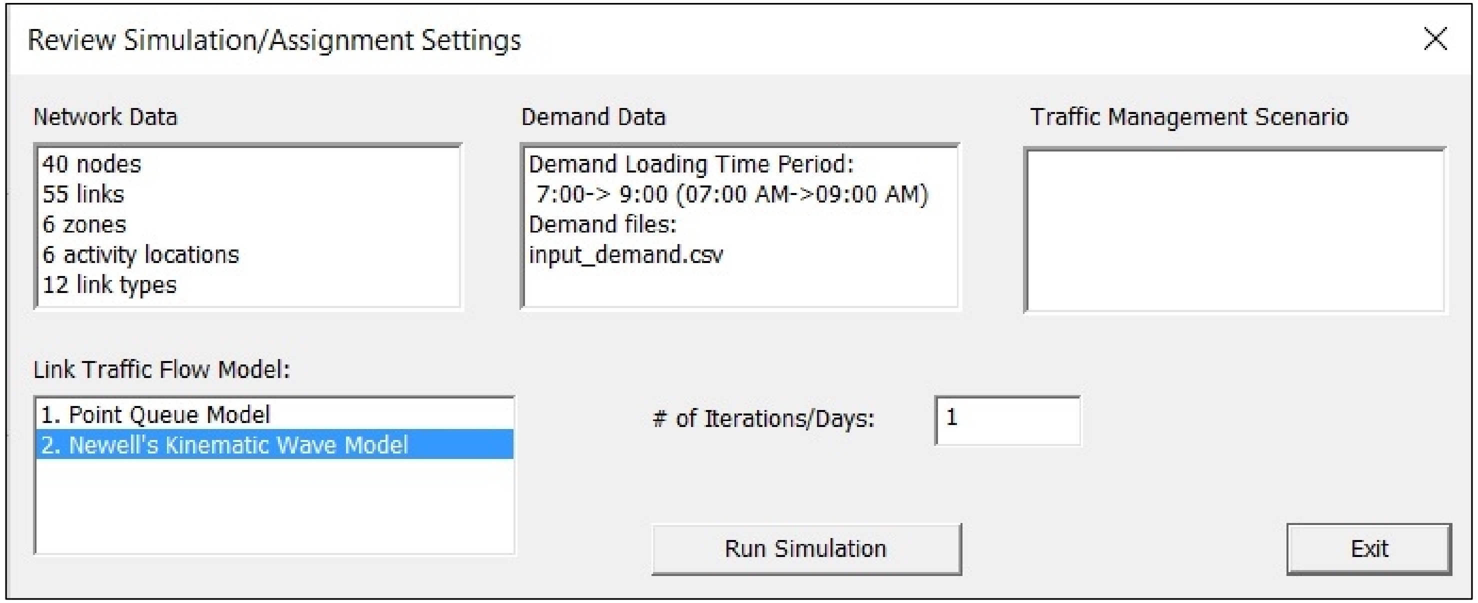
- The traffic demand from origin zone 1 to destination zone 2 is 7200 vehicles in two hours. You can check value of traffic demand from “input_demand.csv” in the project folder.

# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

## Step 2: Setup simulation settings

- Please use the following simulation settings shown in the flowing figure, and run simulation
- Link Traffic Flow Model: “Newell’s Kinematic Wave Model”
- # of Iterations/Days: 1



# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 3: View the result of simulation and analyze traffic states

- View the simulation result: “Total Volume over Capacity Ratio” in the file output_summary.csv

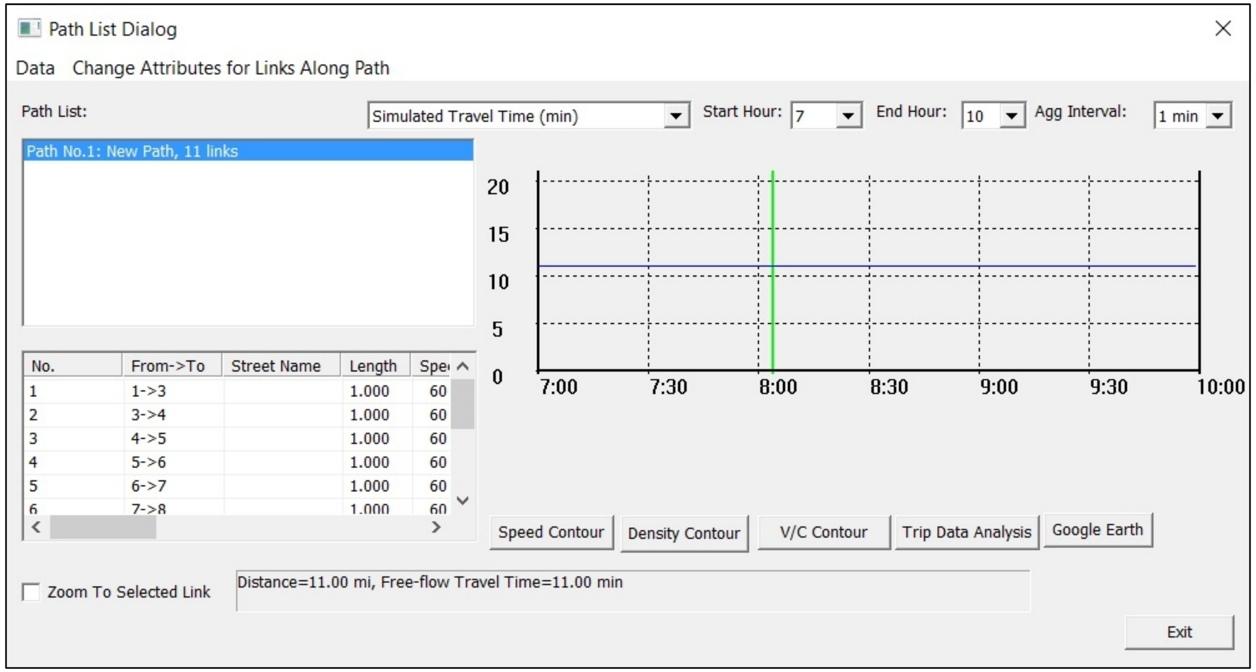
| from_node_id | to_node_id | type_code | link_type | link_length | start_time_in_min | end_time_in_min | total_link_volume | lane_capacity_in_vhc_per_hour | link_capacity_in_vhc_per_hour | volume_over_capacity_ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 3 | f | Freeway | 1 | 420 | 540 | 7200 | 1900 | 5700 | **0.6316** |
| **3** | 4 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **4** | 5 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **5** | 6 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **6** | 7 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **7** | 8 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **8** | 9 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **10** | 11 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 3600 | **1** |
| **11** | 12 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 5400 | **0.6667** |
| **9** | 10 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 3600 | **1** |
| **12** | 2 | f | Freeway | 1 | 420 | 540 | 7200 | 1800 | 7200 | **0.5** |

# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 3: View the traffic states of one path

- Choose a path in the network
- Click on the “Density Contour”, “Speed Contour”, and “V/C Contour” to analyze the details of this traffic assignment.

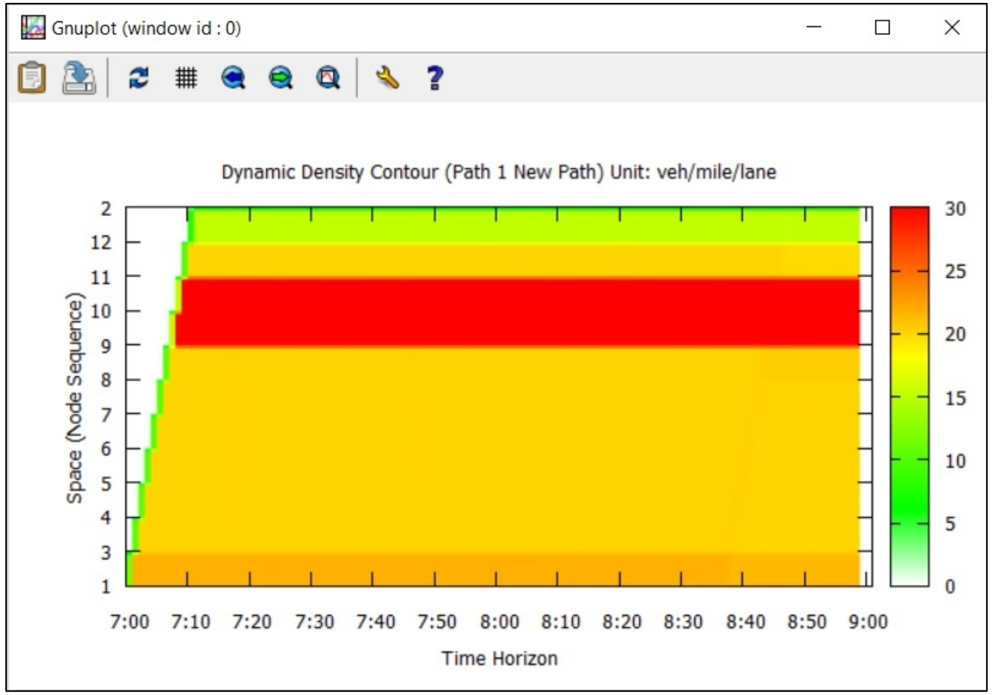


# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 3: View the traffic states of one path

- Similarly, other analyses can also be displayed easily.
- link 9->10 and link 10->11 has a high density, which is consistent with the above “Total Volume over Capacity Ratio”.

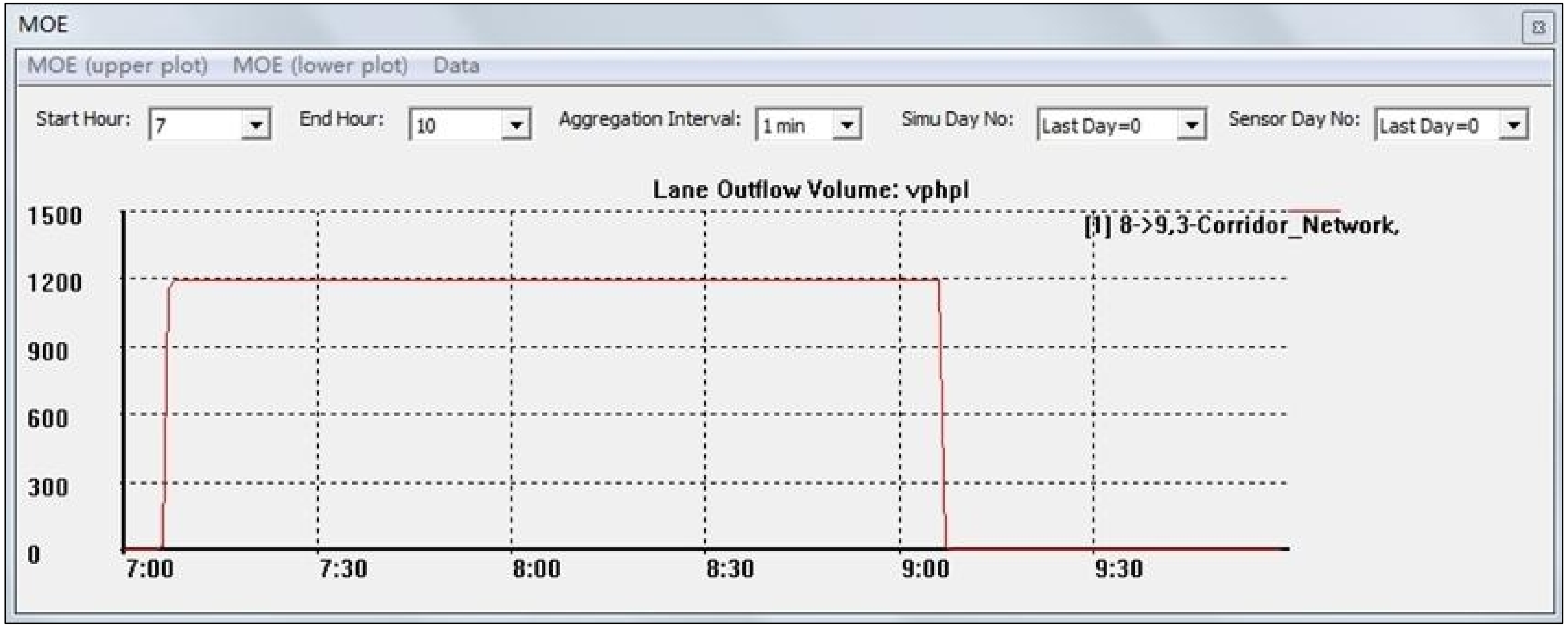


# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

## Step 4: View link MOE of bottleneck links

- Simulated lane volume of link 8->9

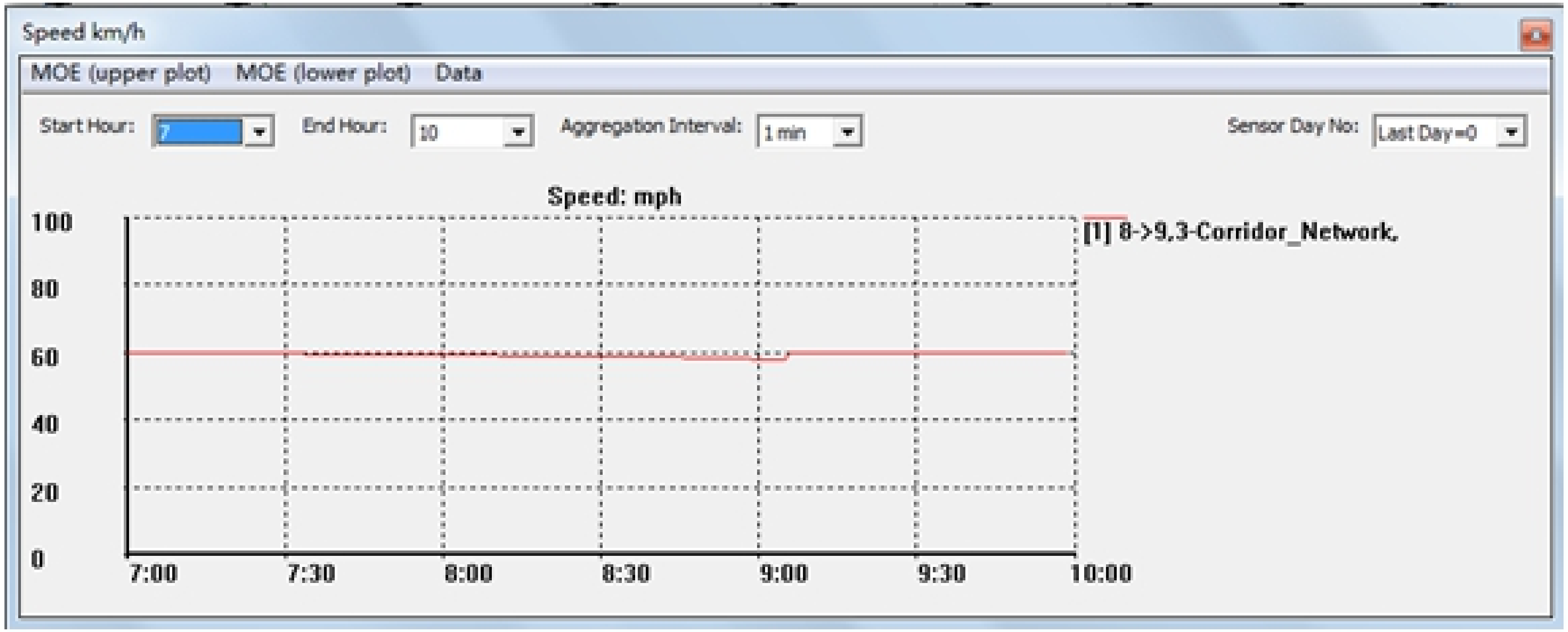


# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 4: View link MOE of bottleneck links

- Simulated speed of link 8->9

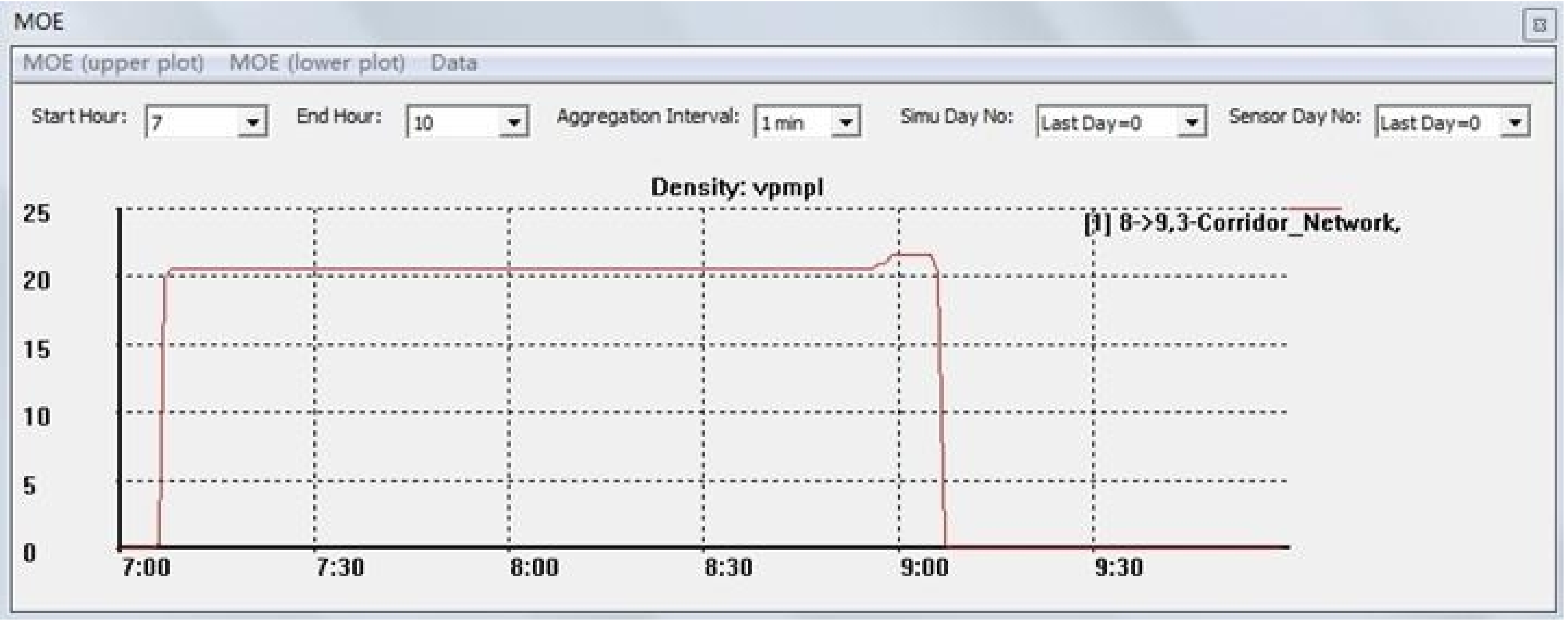


# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Step 4: View link MOE of bottleneck links

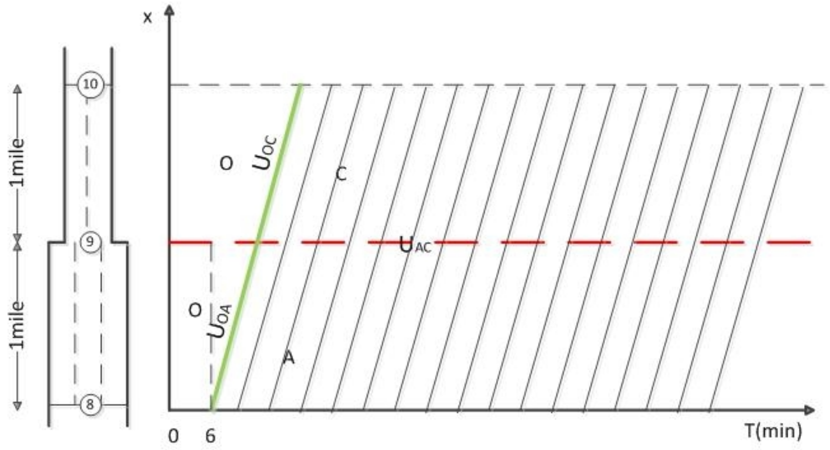
- Simulated density of link 8->9



# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Graphical Method



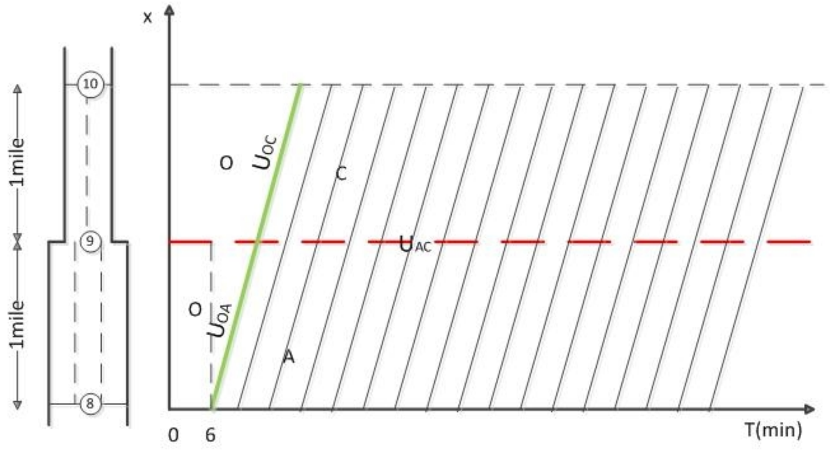

Vehicles' trajectory in time-space diagram

- Speed
- The speed of link 8->9 and link 9->10 is the free flow speed, 60 mph, because the demand is equal to the capacity for the two links.

# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Graphical Method



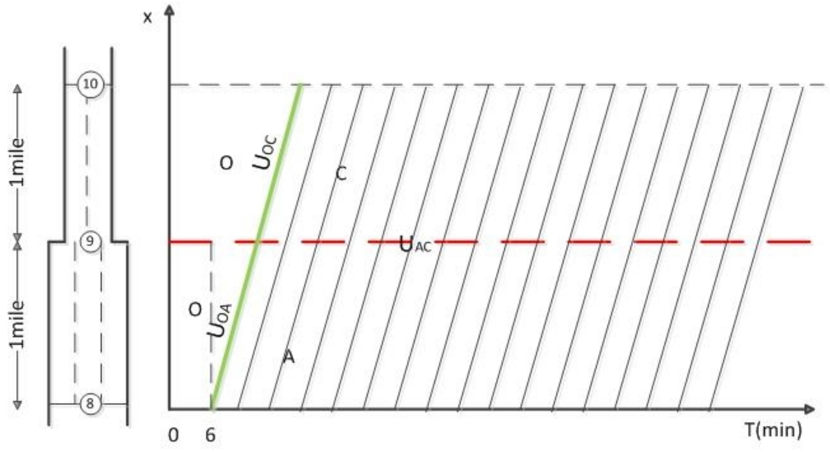

Vehicles' trajectory in time-space diagram

- Density
- Link 8->9 is always at the state A and link 9->10 is at the state C, so its density is 60 vpmpl(link) for the both links.
- Meanwhile, about the density of each lane, link 8->9 is 20 vpmpl(lane) and link 9->10 is 30 vpmpl(lane).

# 5.1 Traffic Congestion Propagation

## Case 1: demand = supply (multiplier = 1.0)

### Graphical Method




Vehicles' trajectory in time-space diagram

- Propagation time: N/A

# 5.1 Traffic Congestion Propagation

**Case 2: demand > supply (multiplier = 1.2)**

**Step 1: Check the value of traffic demand**

- When the demand multiplier is 1.2, the hourly OD demand is 0.5*7200*1.2 (4320).
- The limit capacity of the path 1 is 3600, so now the demand is slightly higher than the path capacity.
- As a result, a traffic congestion is formed and further propagated to the upstream corridor.

# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Step 2: Setup simulation settings

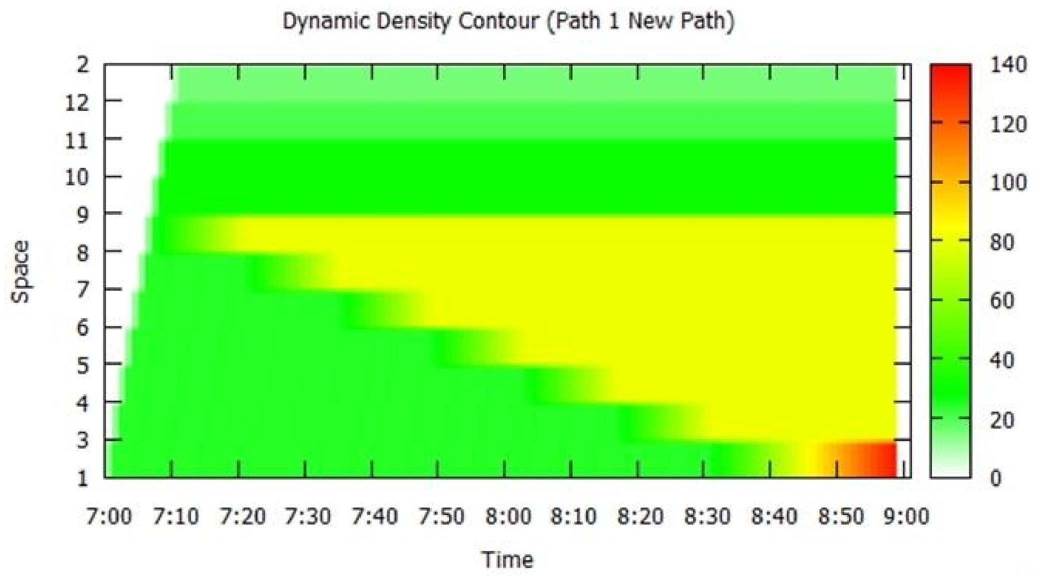
- Go to input_demad_file_list.csv
- Change the loading multiplier to 1.2 in the excel sheet
- Run the simulation for 1 iteration.

# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Step 3: View link MOE of path 1

- traffic congestion starts from link 8->9, and then propagates back to upstream links
- Then the traffic jam reaches the starting link 1->3 (without further space to accommodate the congestion), leading to continuously increasing density

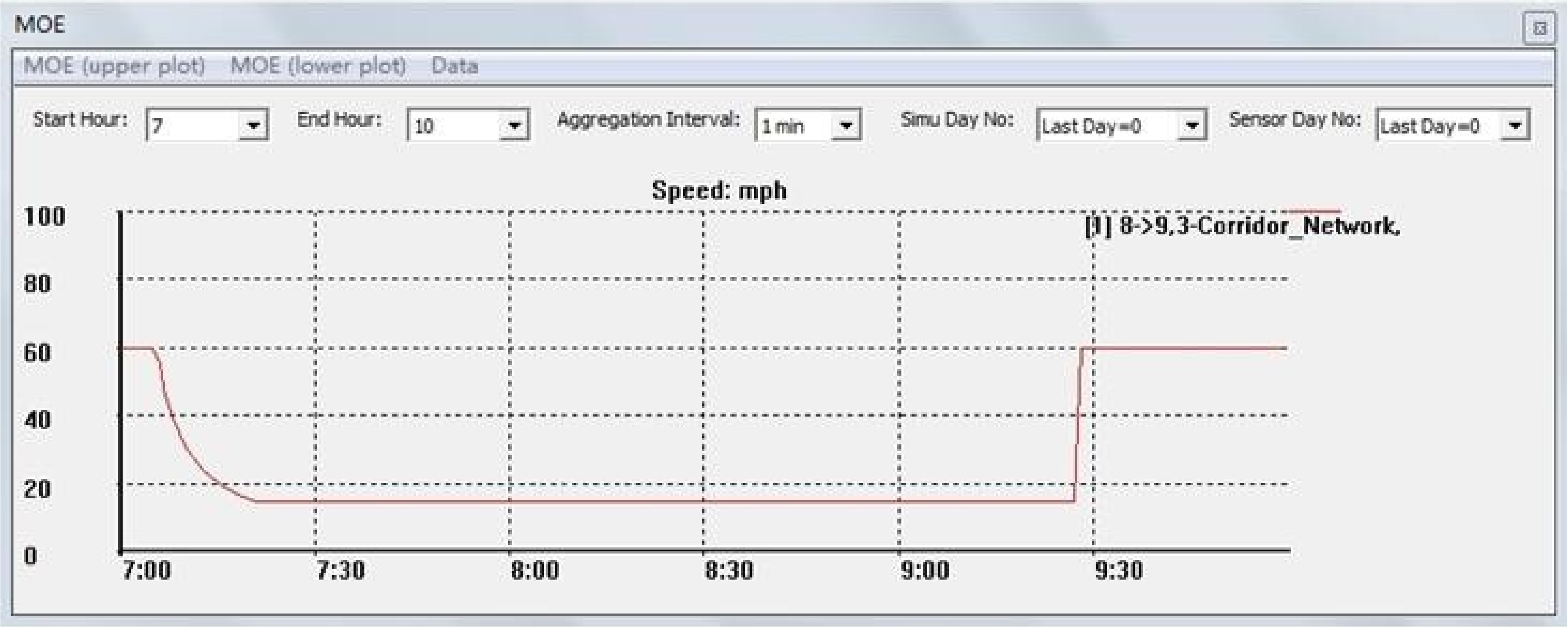


# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

## Step 4: View link MOE of bottleneck links

- Simulated speed of link 8->9

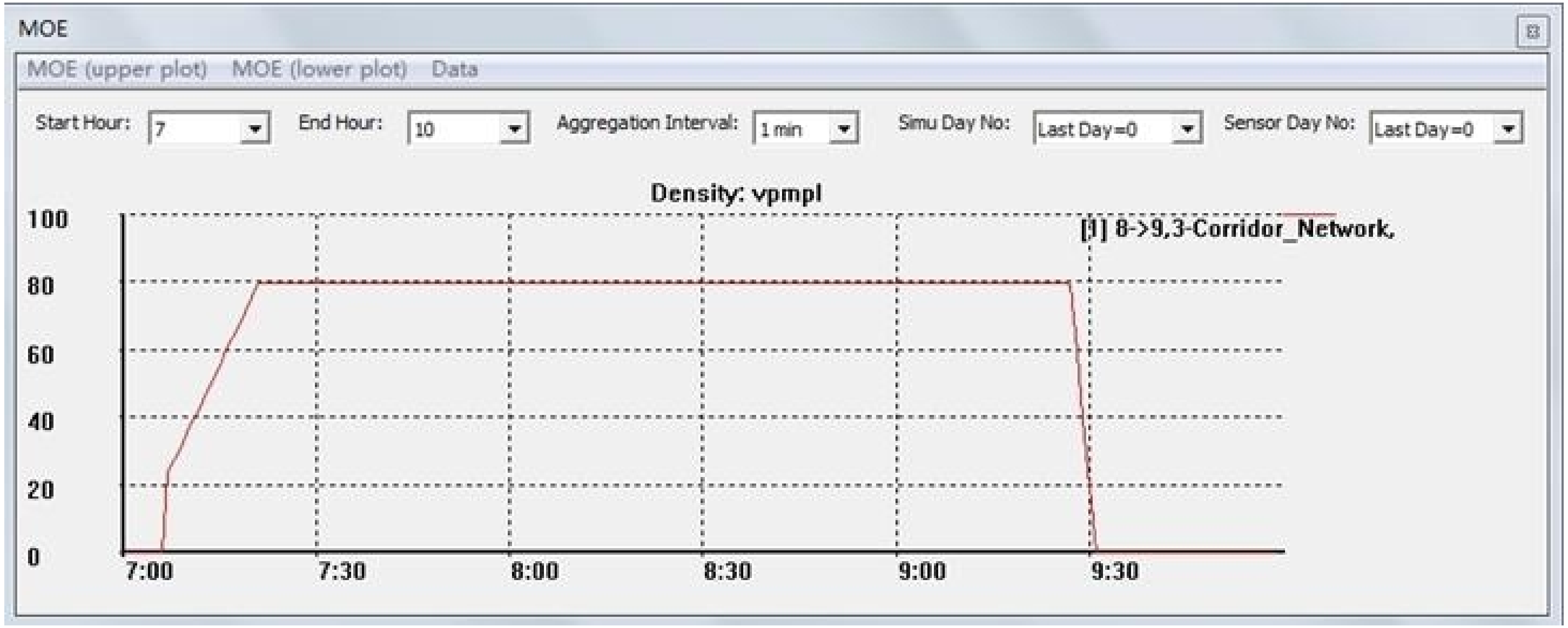


# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Step 4: View link MOE of bottleneck links

- Simulated density of link 8->9



# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

**Step 5: Describe traffic speed and density evolutions on links 7->8 and 8->9, Calculate the traffic congestion speed**

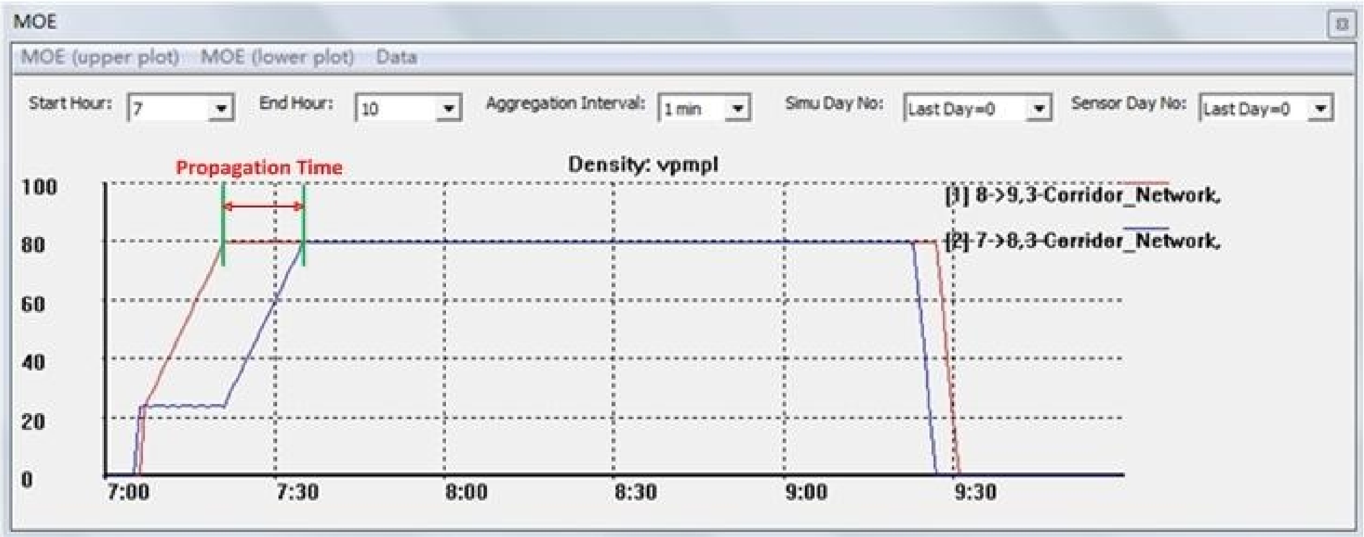
- The speed of link 8->9 changes from 60 mph to 15 mph in the first 20 min, and its density increases from 24 vpmpl to 80 vpmpl.
- To calculate traffic congestion propagation speed, we need to measure the propagation time and link length (1 mile)

# 5.1 Traffic Congestion Propagation

**Case 2: demand > supply (multiplier = 1.2)**

**Step 5: Describe traffic speed and density evolutions on links 7->8 and 8->9, Calculate the traffic congestion speed**

- Select two links simultaneously, link 8->9 and link 7->8.
- Using key combination “ctrl +mouse click”, we can obtain the link MOE plot and then measure the queue propagation time.



# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

**Step 5: Describe traffic speed and density evolutions on links 7->8 and 8->9, Calculate the traffic congestion speed**

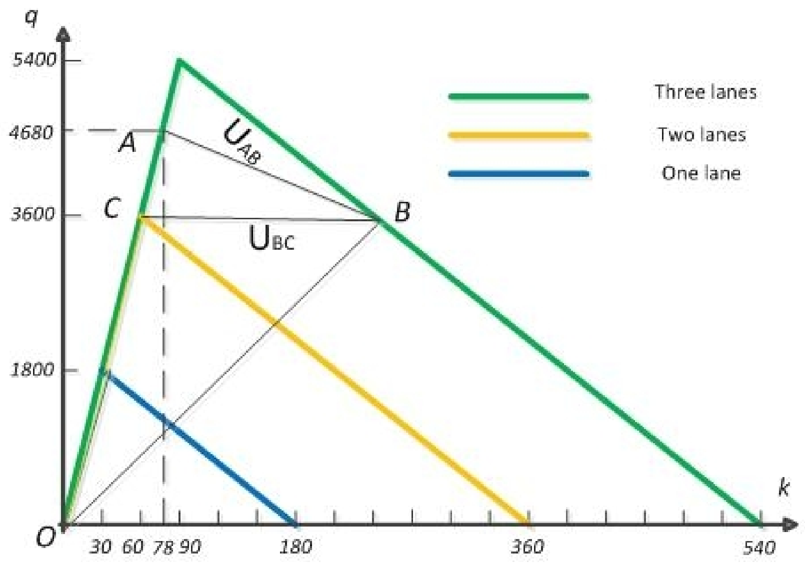
- One can also precisely measure the exact propagation time duration using exported link MOE csv file.

- When the density (link 8->9) becomes 80, the corresponding time is 441min.
- When the density (link 7->8) becomes 80, the corresponding time is 455 min.
- Thus, the propagation time duration is 14 min.

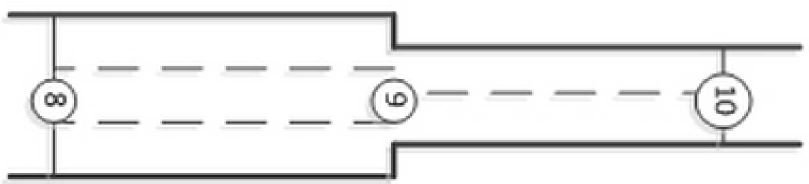
# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Graphical Method




The flow-density relationship for different states of links

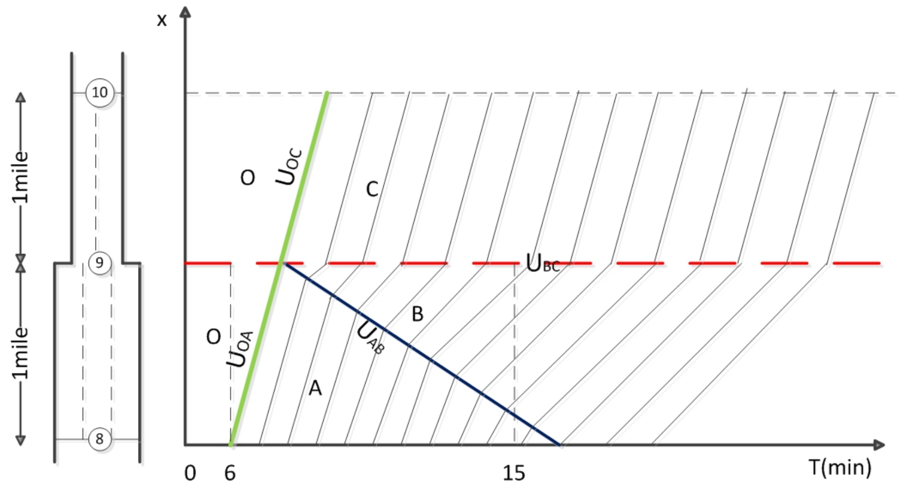



- U: interface between two different traffic states.
- State O: there is no traffic flow
- State A: the demand traffic enter the link 8->9
- State B: the steady traffic state of link 8->9
- State C: the steady traffic state of link 9->10

# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Graphical Method



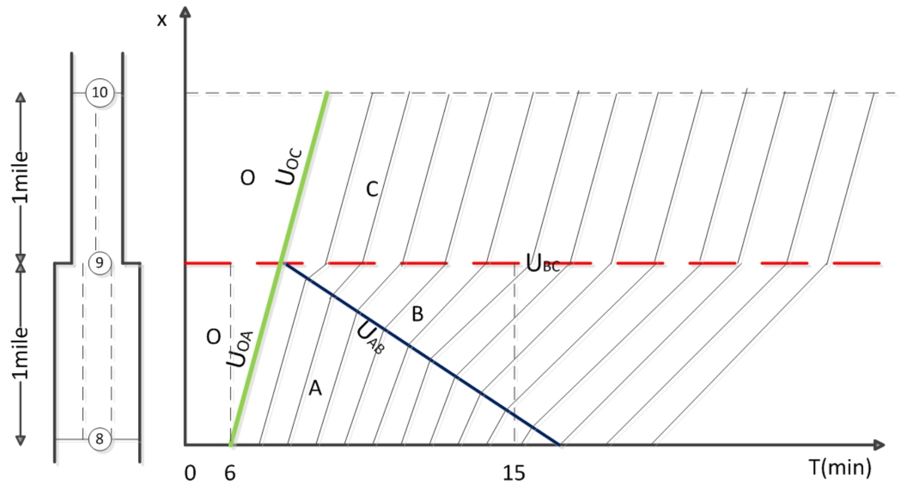
Vehicles' trajectory in time-space diagram

- Speed
- for link 8->9, at state A, its speed is 60 mph; at state B, due to the limitation of capacity, its speed is 15 mph.
- For link 9->10, the vehicles will drive at a speed based on the capacity, state C, whose speed is 60 mph;

# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Graphical Method



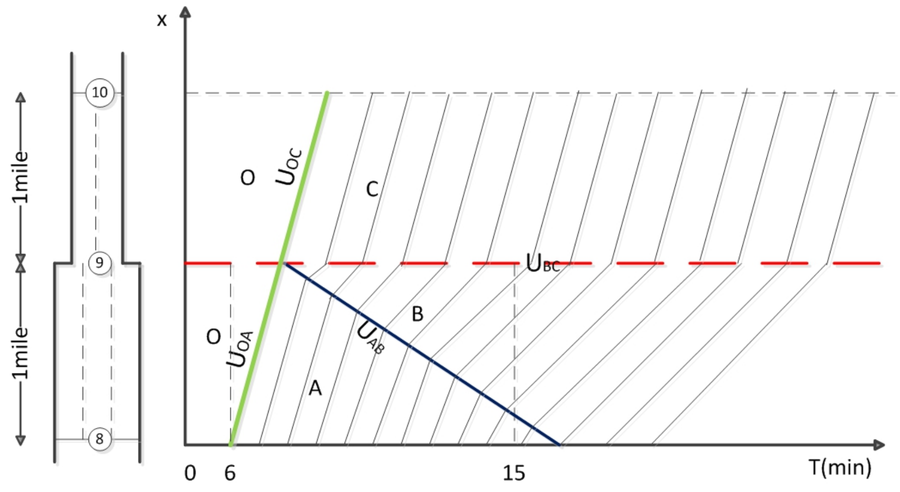

Vehicles' trajectory in time-space diagram

- Density
- for link 8->9, at the state A, its density is 72 vpmpl(link). After the propagation time, it is at the state B, whose density is 240 vpmpl(link).
- Meanwhile, the link 9->10 is always at the state C, whose density is 60 vpmpl(link).

# 5.1 Traffic Congestion Propagation

## Case 2: demand > supply (multiplier = 1.2)

### Graphical Method




Vehicles' trajectory in time-space diagram

- Propagation time: 14 min.

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Step 1: Check the value of traffic demand

- In this case, the demand OD is still 7200 from 7am to 9am,
- the multiplier is 1.3. so now the demand is much higher than the path capacity.
- The analysis process is the same as case 2 (0.5*7200*1.3 = 4680 vehicles).

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Step 2: Setup simulation settings

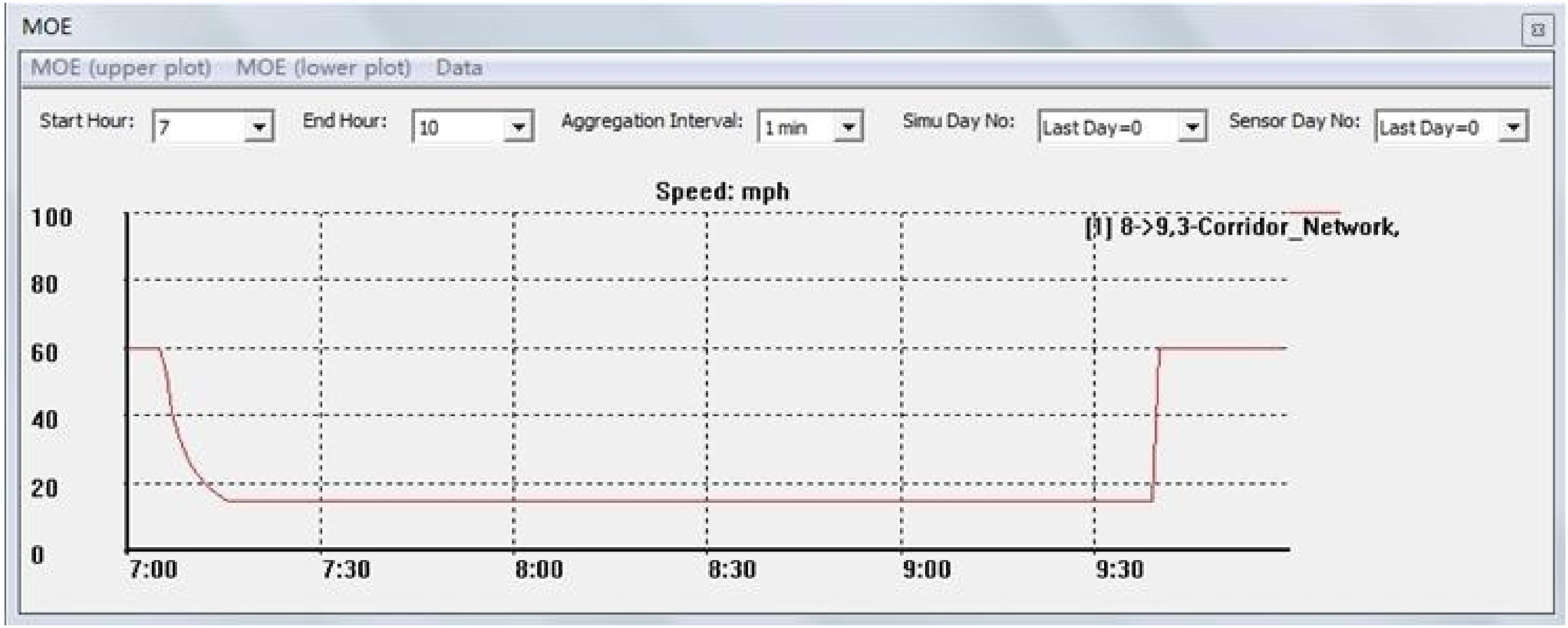
- Go to input_demad_file_list.csv
- Change the loading multiplier to 1.3 in the excel sheet
- Run the simulation for 1 iteration.

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Step 3: View link MOE of bottleneck links

- Simulated speed of link 8->9

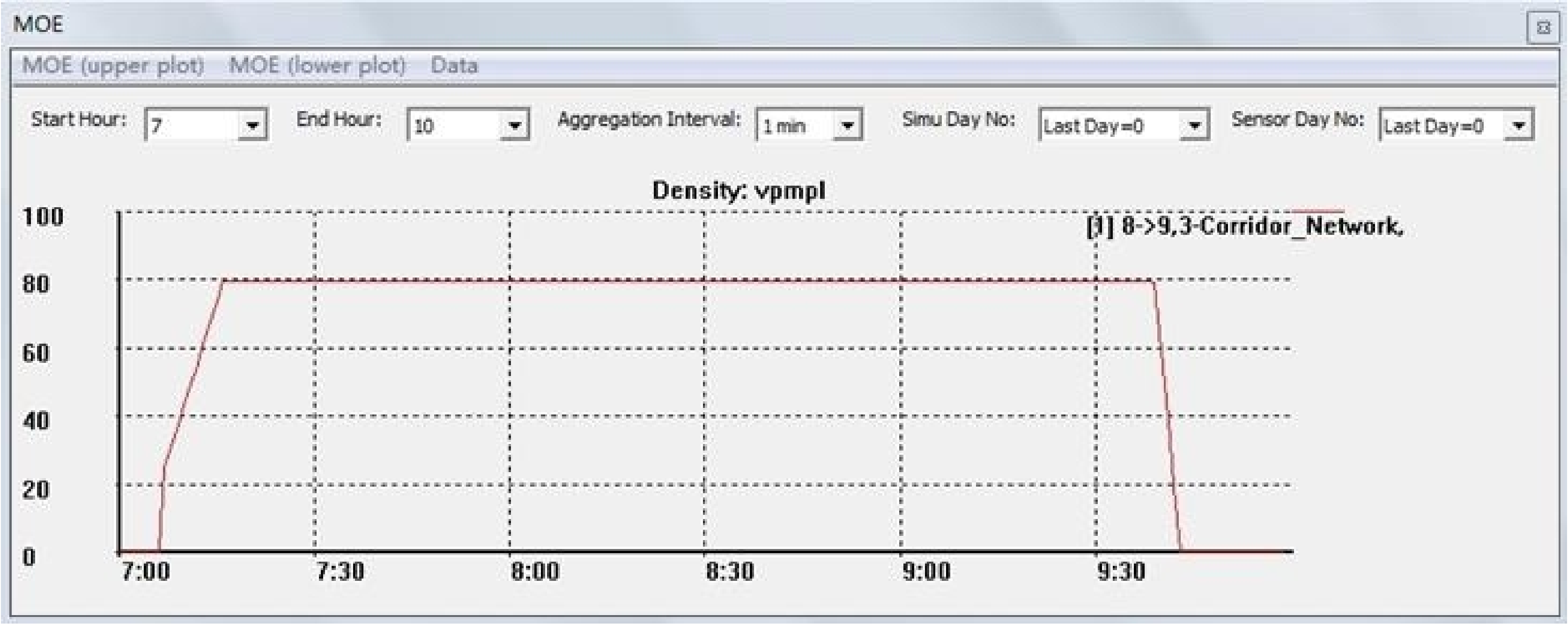


# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Step 3: View link MOE of bottleneck links

- Simulated density of link 8->9

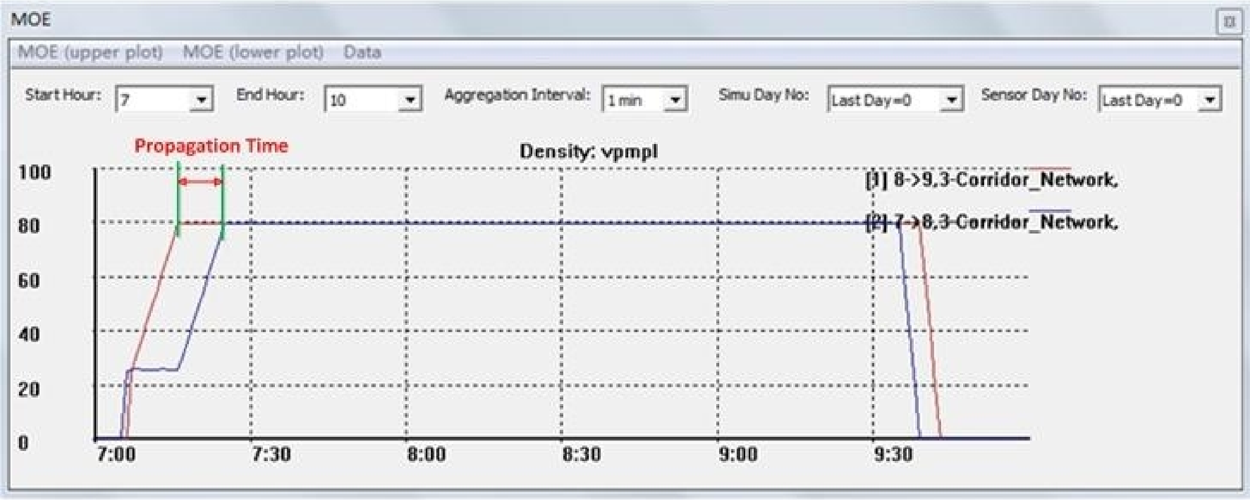


# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Step 4: Describe traffic speed and density evolutions on links 7->8 and 8->9, Calculate the traffic congestion speed

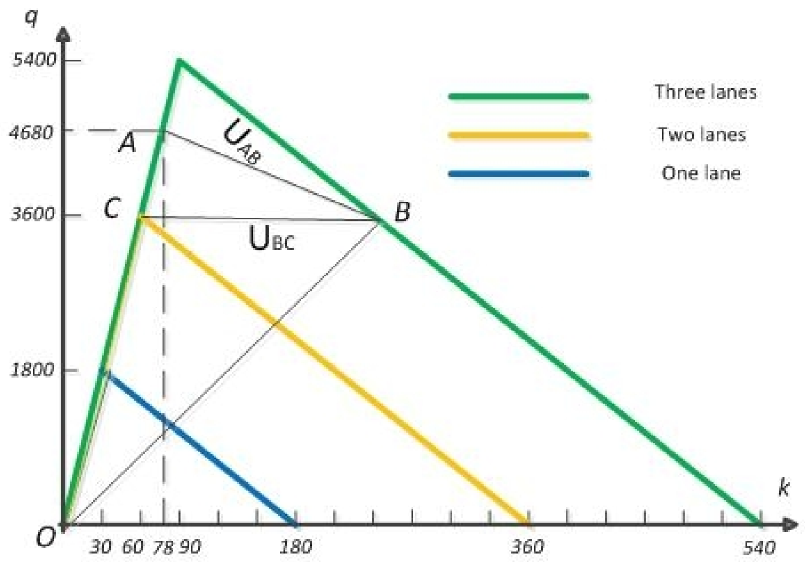
- The speed of link 8->9 changes from 60 mph to 15 mph, and its density of one lane increases from 26 vpmpl to 80 vpmpl.
- A traffic congestion propagation time of 9 min can be observed in the following figure.



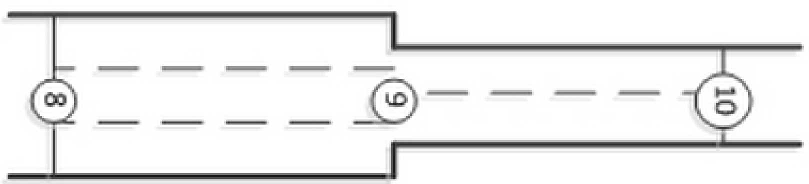
# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Graphical Method




The flow-density relationship for different states of links

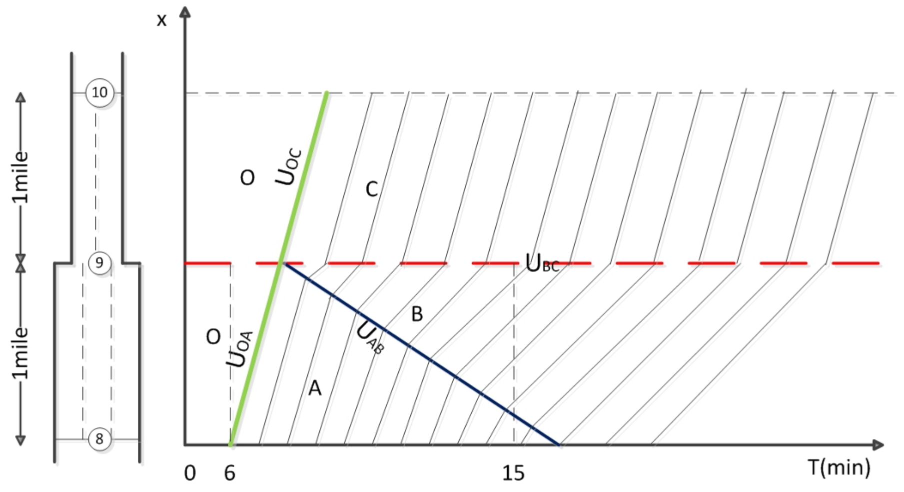



- U: interface between two different traffic states.
- State O: there is no traffic flow
- State A: the demand traffic enter the link 8->9
- State B: the steady traffic state of link 8->9
- State C: the steady traffic state of link 9->10

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Graphical Method



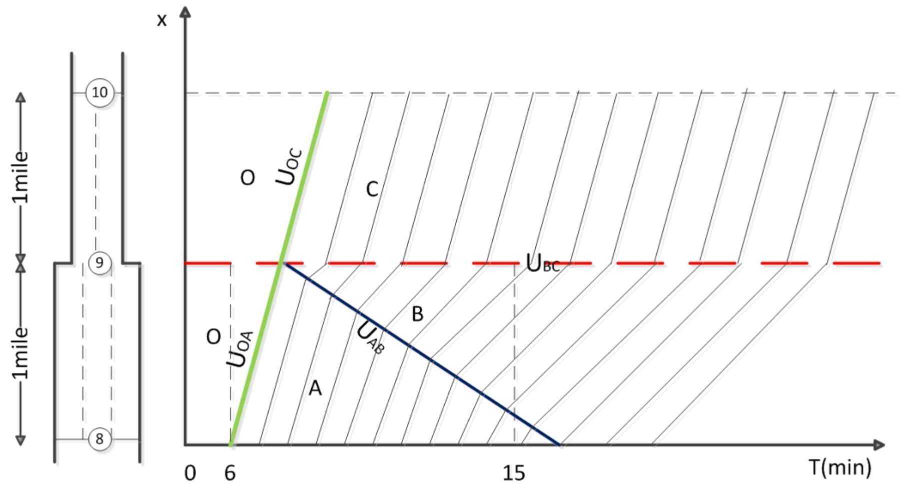

Vehicles' trajectory in time-space diagram

- Speed
- For link 8->9, at state A, its speed is 60 mph; at state B, due to the limitation of capacity, its speed is 15 mph.
- For link 9->10, the vehicles will drive in a speed based on the capacity, state C, whose speed is 60 mph;

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Graphical Method



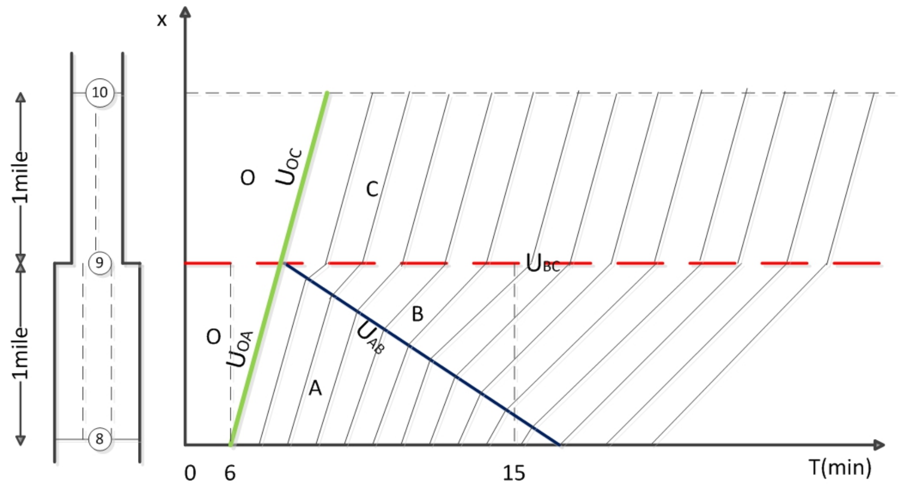

Vehicles' trajectory in time-space diagram

- Density
- For link 8->9, at state A, its density is 78 vpmpl(link). After the propagation time, at state B, its density is 240 vpmpl.
- Meanwhile, the link 9->10 is always at the state C, whose density is 60 vpmpl(link). In addition, the method of calculating the density of each lane is the same as case 1.

# 5.1 Traffic Congestion Propagation

## Case 3: demand > supply (multiplier = 1.3)

### Graphical Method

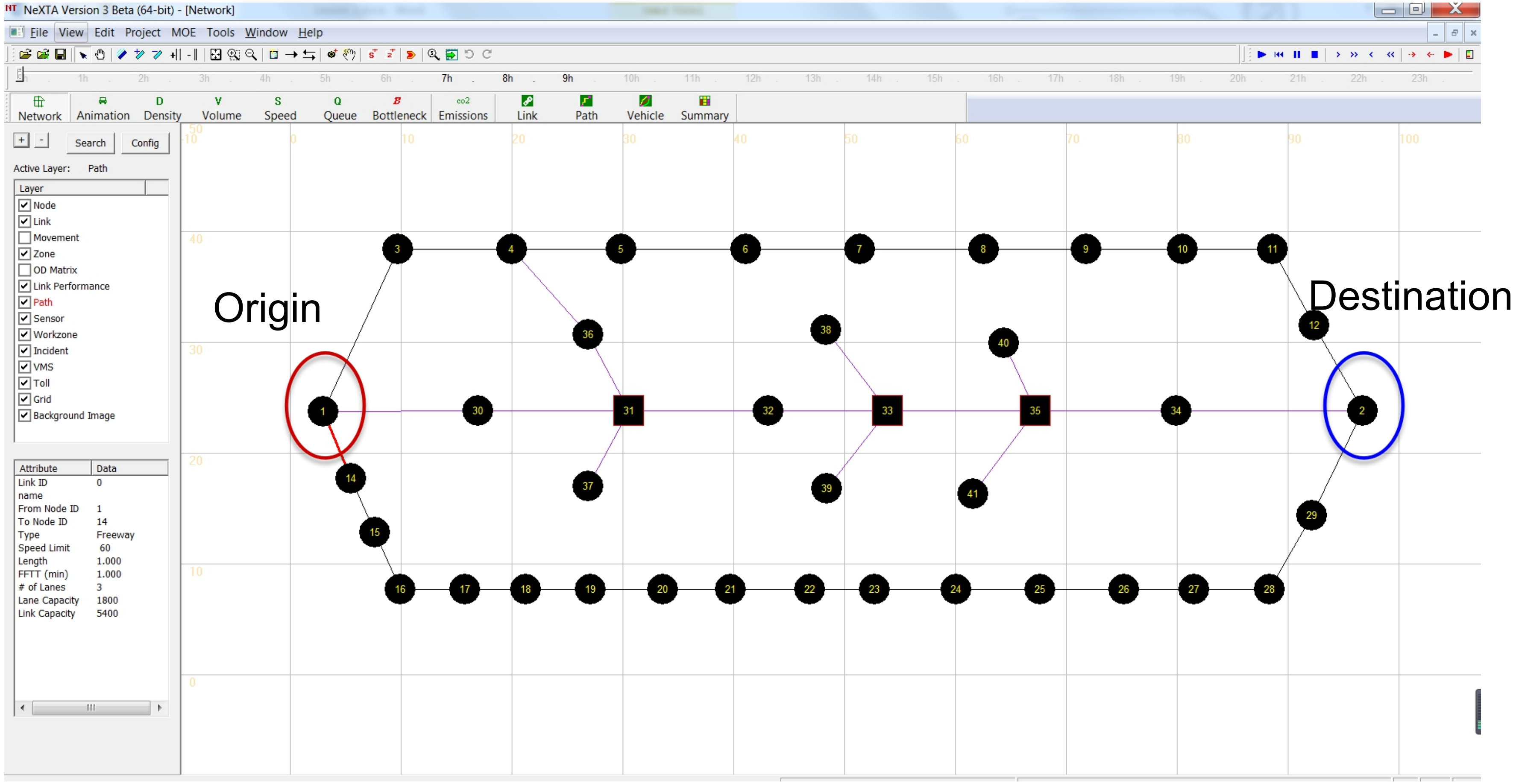


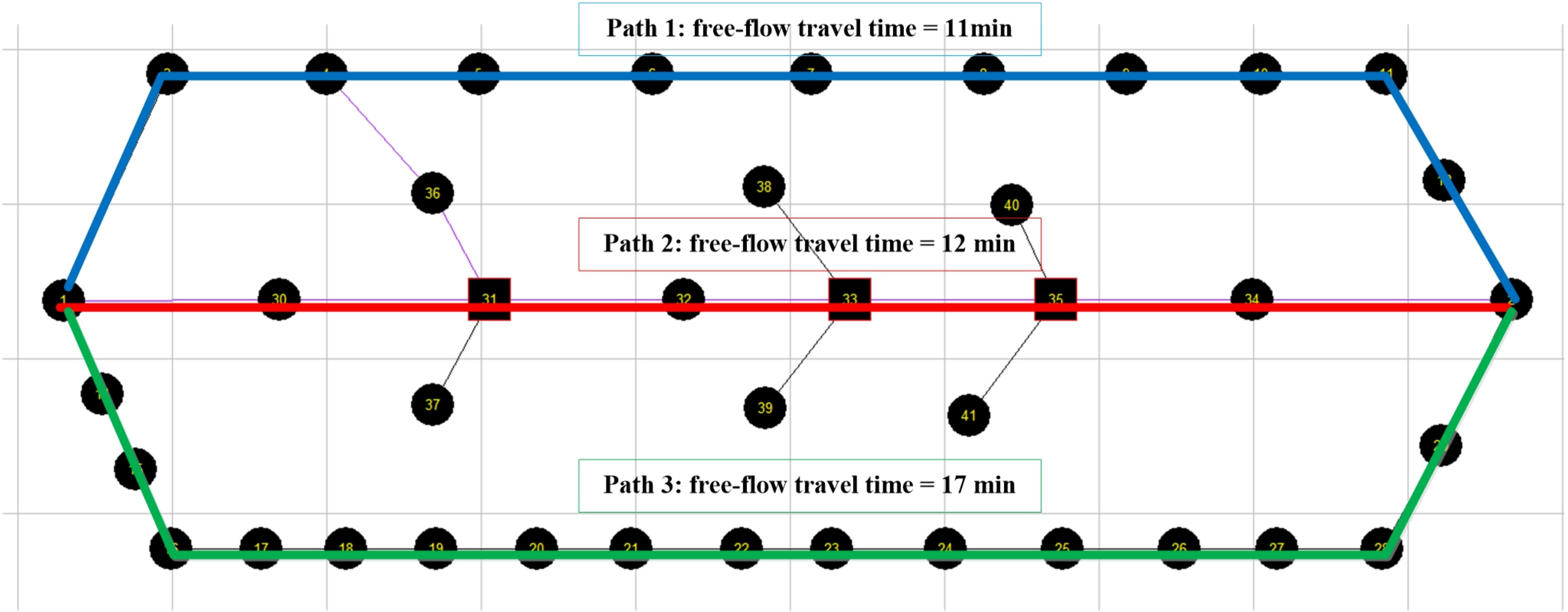

Vehicles' trajectory in time-space diagram

- Propagation time: 9 min.

# 5.2 Traffic States as a result of demand-supply interactions

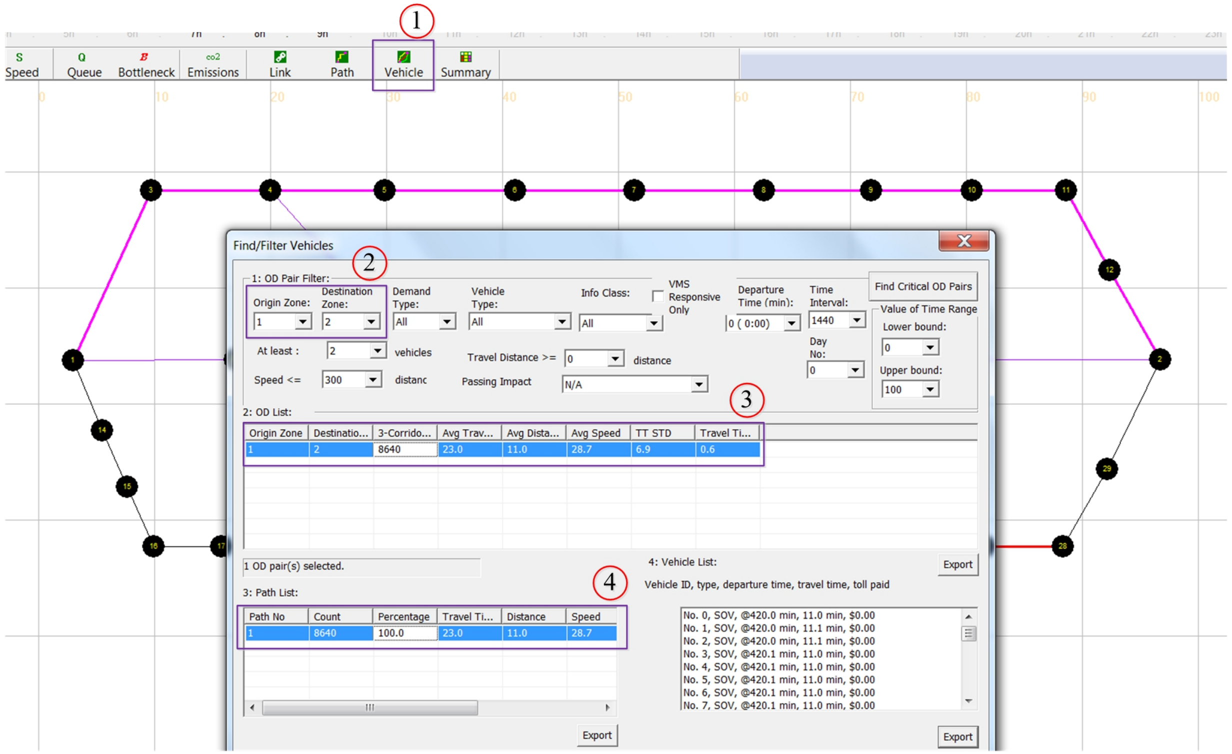
- Tolling example: 3-corridor network





Path 1: free-flow travel time = 11min
Path 2: free-flow travel time = 12 min
Path 3: free-flow travel time = 17 min
30
31
32
33
35
34
36
37
38
39
40
41
17
18
19
20
21
22
23
24
25
26
27

- Run 1 iteration: all vehicles choose the least-cost path based on link free-flow travel time.




Step 1: Click "Vehicle" to call the dialogue of "Find/Filter Vehicle".

Step 2: Choose the target OD pair and corresponding conditions. In this case, we choose OD pair 1→2.

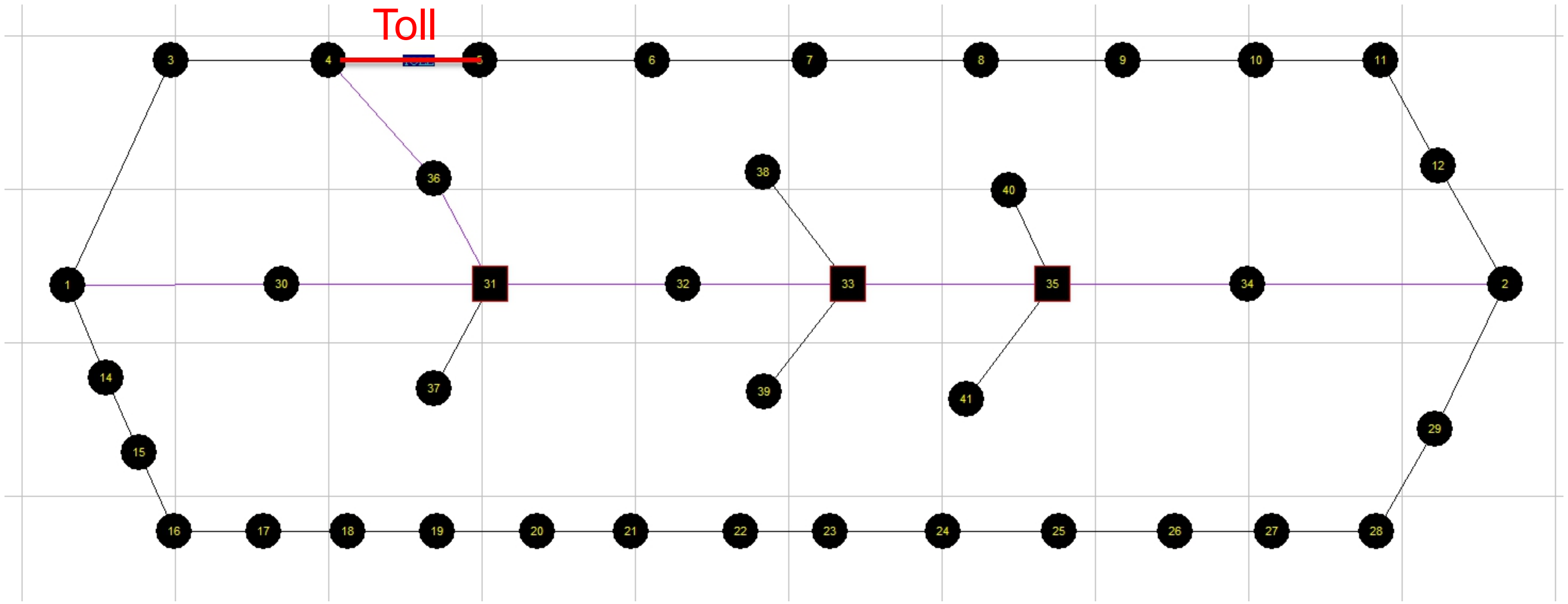
Step 3: Click the OD list to check the general summary of OD pair 1→2.

Step 4: Choose one used path in Path List

- Individual generalized cost function

$$Cost = Travel\,Time * VOT + Toll$$

  - Can consider multiple factors
    - Value of time, Value of reliability, Value of safety
  - Perform routing algorithm individually for each vehicle/agent
  - Can adjust origin/destination/departure time/path at each iteration (day)




The specific toll settings are **input in Scenario_Link_Based_Toll.csv**.

| Link | Scenario No | Start Day | End Day | Start Time in Min | End Time in min | Toll for Demand Type 1 |
|---|---|---|---|---|---|---|
| [4,5] | 0 | 0 | 1 | 420 | 540 | 0.2 |

# link-based toll for low-income travelers

Average VOT of SOV in the file **input_demand_type.csv**

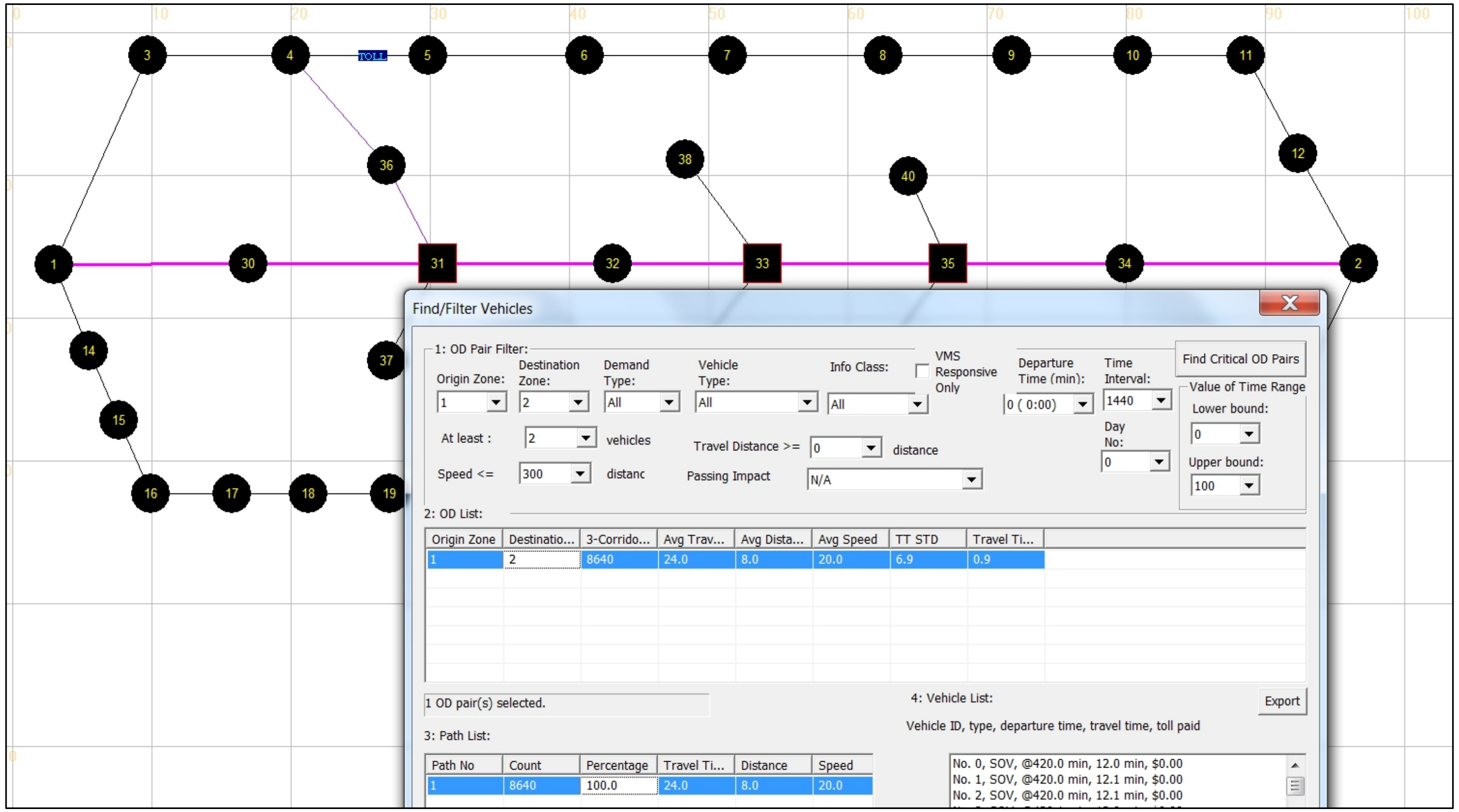
| demand_type | demand_type_name | avg_VOT |
|---|---|---|
| 1 | SOV | 10 |

The generalized link travel time can be formulated as $GC_{i,j} = t_{i,j} + \frac{toll_{i,j}}{VOT}$

Path 1 with toll values will have the generalized travel time value of $\left(11 + \frac{0.2}{10/60}\right)$ = 12.2 min, which is more that the free-flow path travel time of path 2 (12 min).

link-based toll for low-income travelers

- Vehicles will choose path 2

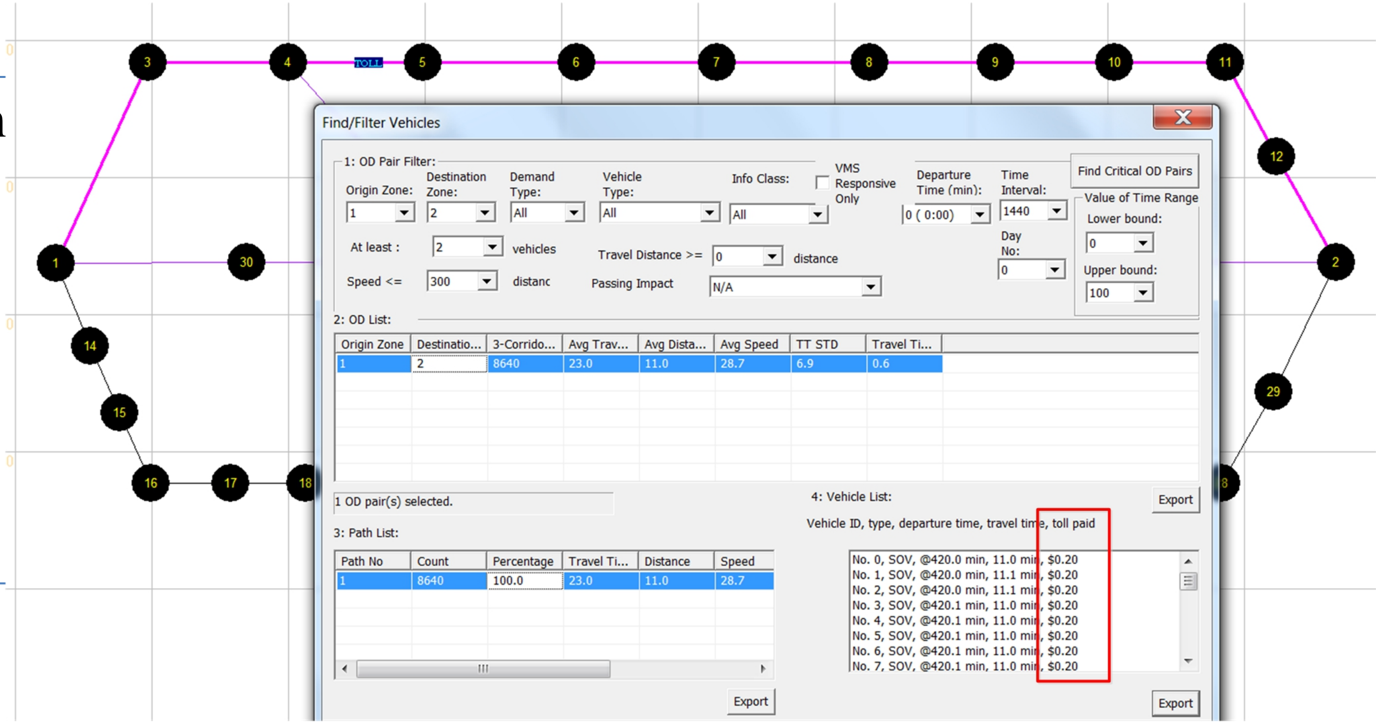


# link-based toll for high-income travelers

Average VOT of SOV in the file input_demand_type.csv

| demand_type | demand_type_name | avg_VOT |
|---|---|---|
| 1 | SOV | 20 |

The generalized path travel time of path 1 will be $11+\frac{0.2}{20/60}=$ 11.6 min, which is less than 12 min of path 2’s free-flow travel time

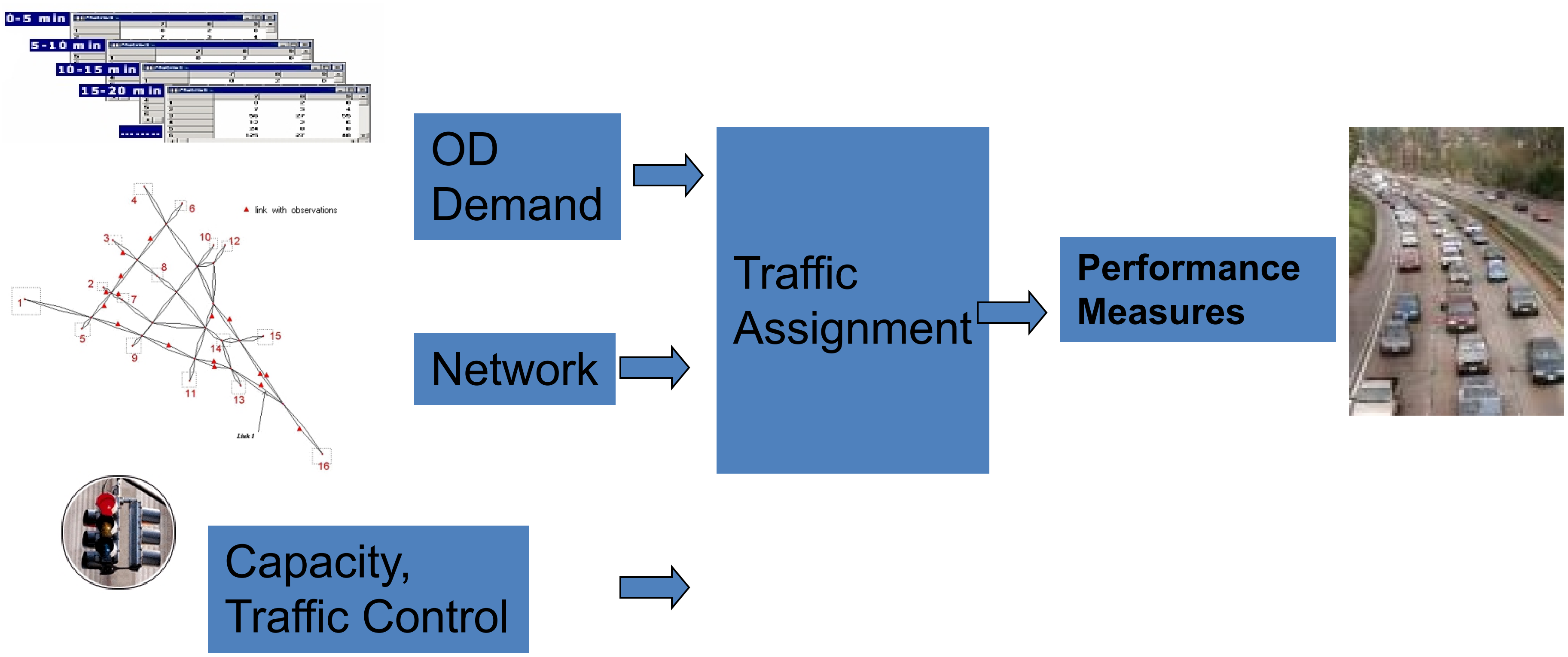


# Module 6
# Origin Demand Matrix Estimation

6.0 Background information

# Why is OD Demand Estimation Important?

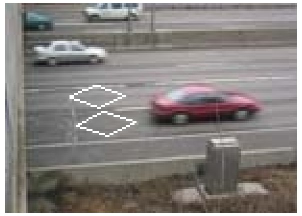
- Essential input for traffic assignment systems to support transportation network planning and operational decisions



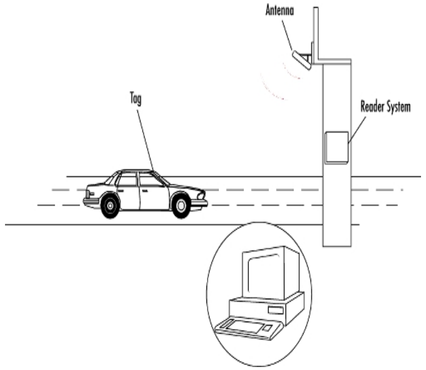
# Why do We Need to Utilize Different Data Sources?

- Advanced sensor technologies offer more reliable and less costly channels to provide real-time traffic flow measurements

Survey Data — Point Sensor — Point-to-point Sensor



Loop Detector




Automatic Vehicle Identification

# How has OD Matrix Estimation been Done?

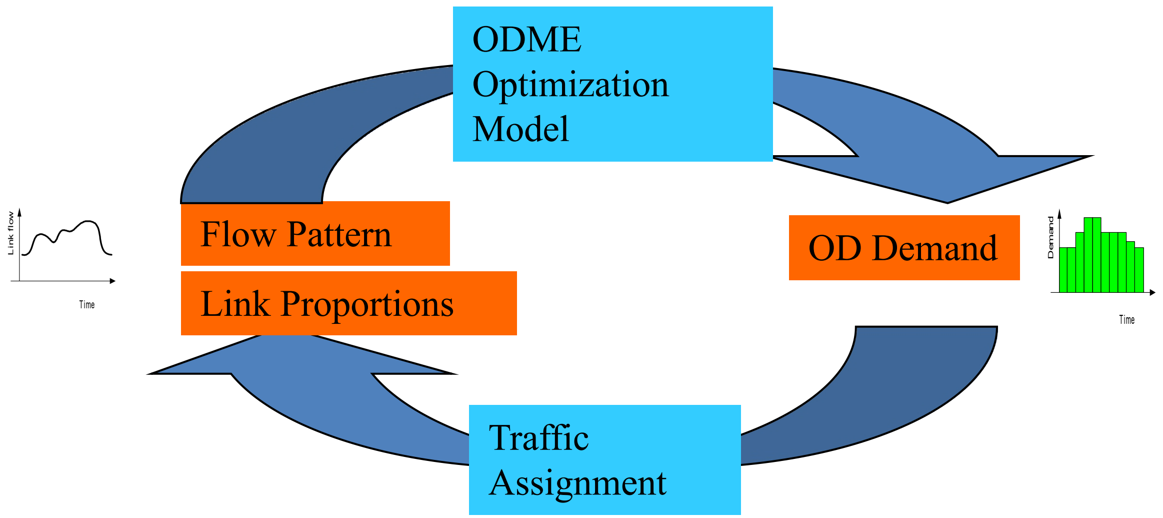
Upper level:

Constrained ordinary least-squares problem

$$\min Z = \sum_{l \in L_{lc}, t} [\sum_{o,d,\tau} \hat{p}_{(l,t),(o,d,\tau)} \cdot D_{(o,d,\tau)} - c_{(l,t)}]^2$$

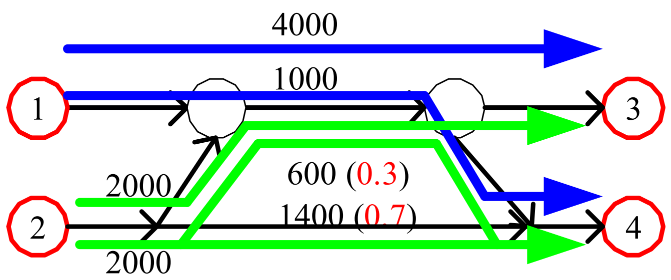
Lower level:

– Link flow proportion $\hat{p}_{(l,h),(o,d,\tau)}$ Traffic assignment ( $D_{(o,d,\tau)}$



# What is Link-Flow Proportion?

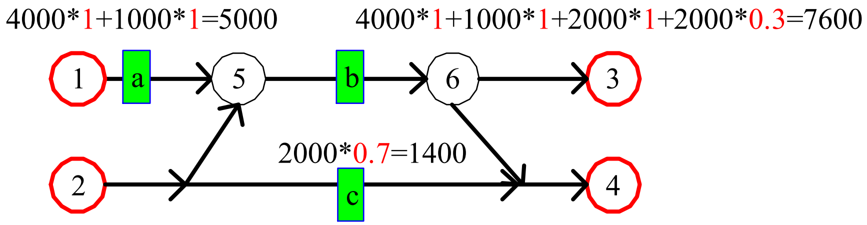
Path Flow Structure




Link Proportions (*Assignment matrix*)
Fraction of vehicular demand flow from **OD pair( *i, j*)** contributing to the flow on **link *l***

$$p_{l,(o,d)} = \frac{c_{(o,d),l}}{D_{o,d}}$$

| OD pair | 1->3 | 1->4 | 2->3 | 2->4 |
|---|---|---|---|---|
| Link a | 1 | 1 | 0 | 0 |
| Link b | 1 | 1 | 1 | 0.3 |
| Link c | 0 | 0 | 0 | 0.7 |



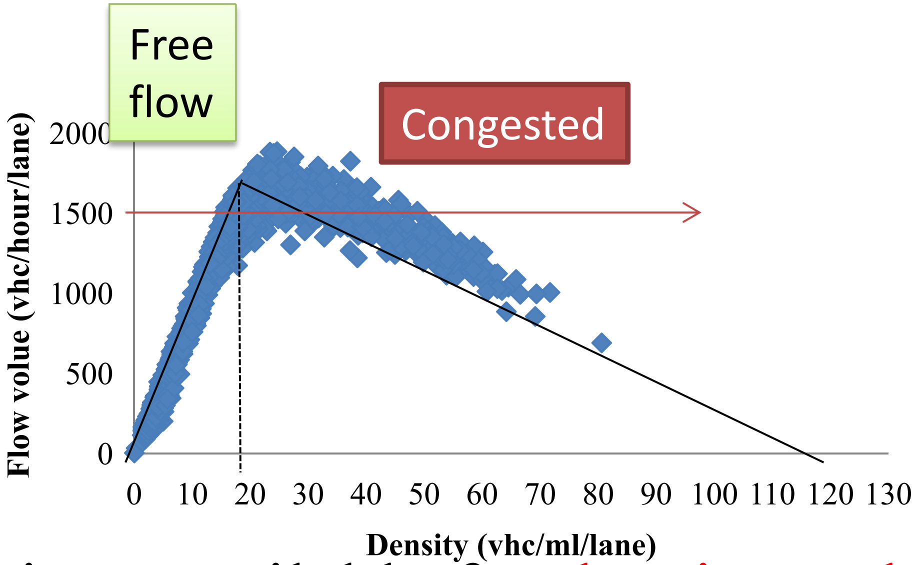
# Two Common Questions…

1. Travel time IS NOT the only factor in travel route choice utility function.
   - A long list of additional route choice factors: Highway bias, distance consideration, # of signals, inherence preferences
   - Do we need to reach the "perfect" travel-time-based UE in ODME?

2. Can we combine/integrate traffic assignment and OD demand estimation? (Both aim to adjust flows in a network)

# What is the Difficulty of Dynamic ODME under Congested Conditions?

Single value of flow volume observation could correspond to free-flow and congested states




No linear link proportions available for density and travel time measurements

# Our Proposed Approaches

1. Path Flow Adjustment
   - Combine  OD estimation/adjustment with traffic assignment
   - Final solution is a set of path flow patterns satisfying “tolled user equilibrium”

2. Approximate Gradient Method
   – Use spatial queue model to calculate link flow/density change due to incoming path flow change

# Objective Function: Minimize

(1) Deviation between measured and estimated traffic states

(2) Deviation between aggregated path flows and target OD flows

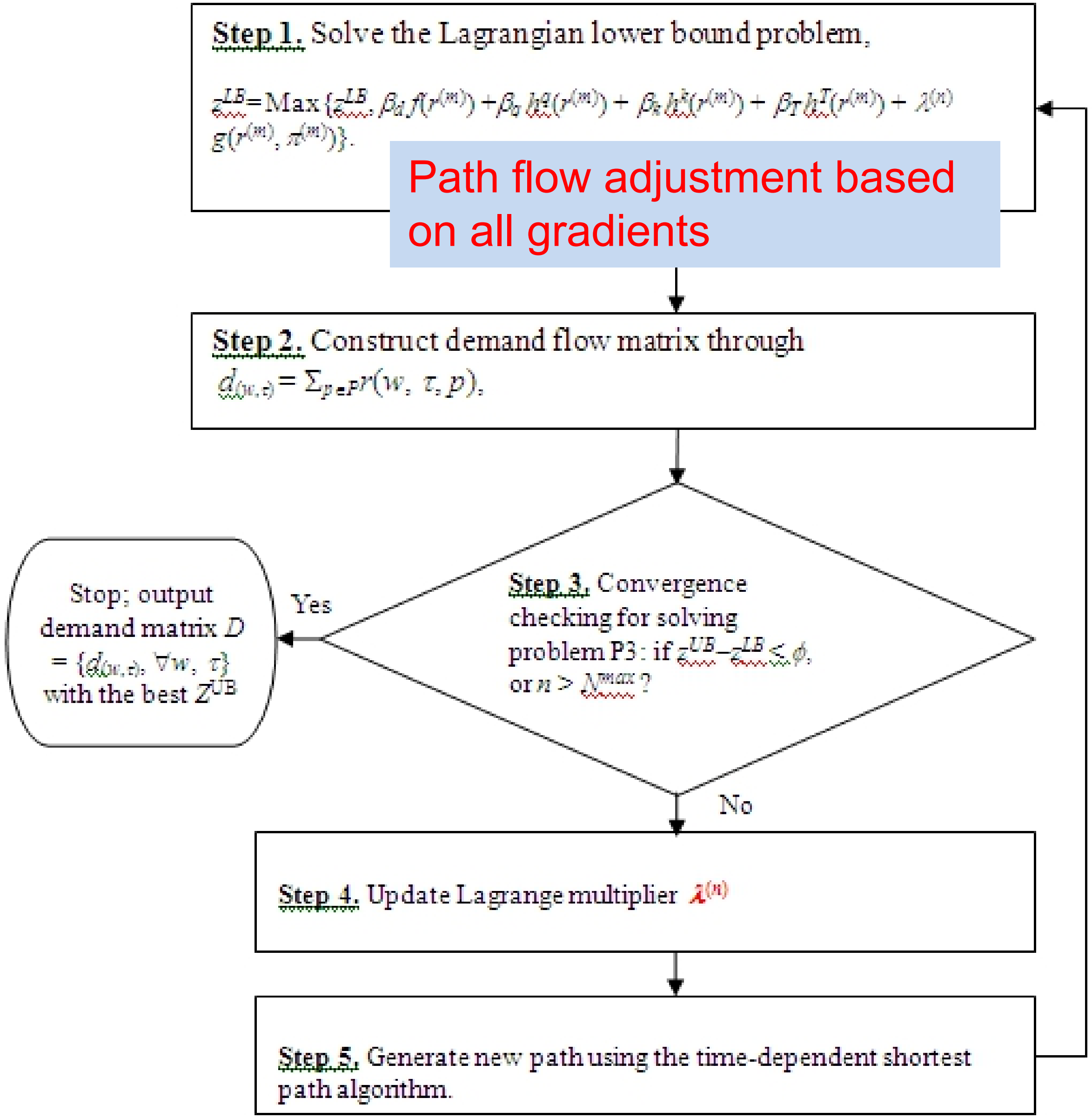
$$\sum_{o \in O} \sum_{d \in D} \sum_{\tau \in T} \left[ D_{od}^{\tau} - \overline{D}_{od}^{\tau} \right] = \sum_{o \in O} \sum_{d \in D} \sum_{\tau \in T} \left[ \sum_{p \in P(o,d,\tau)} r_{odp}^{\tau} - \overline{D}_{od}^{\tau} \right]$$

Demand flow     target demand     path flow     target demand

(3) Gap Function to Quantify User Equilibrium Conditions

$$Gap(r,\pi) = \sum_{o \in O} \sum_{d \in D} \sum_{\tau \in T} \sum_{p \in P(o,d,\tau)} r_{odp}^{\tau} [T_{odp}^{\tau}(r) - \pi_{od}^{\tau}]$$

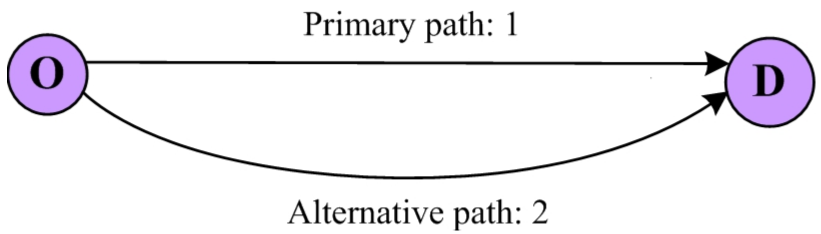
# Flow Chart of Algorithm



# Example 1

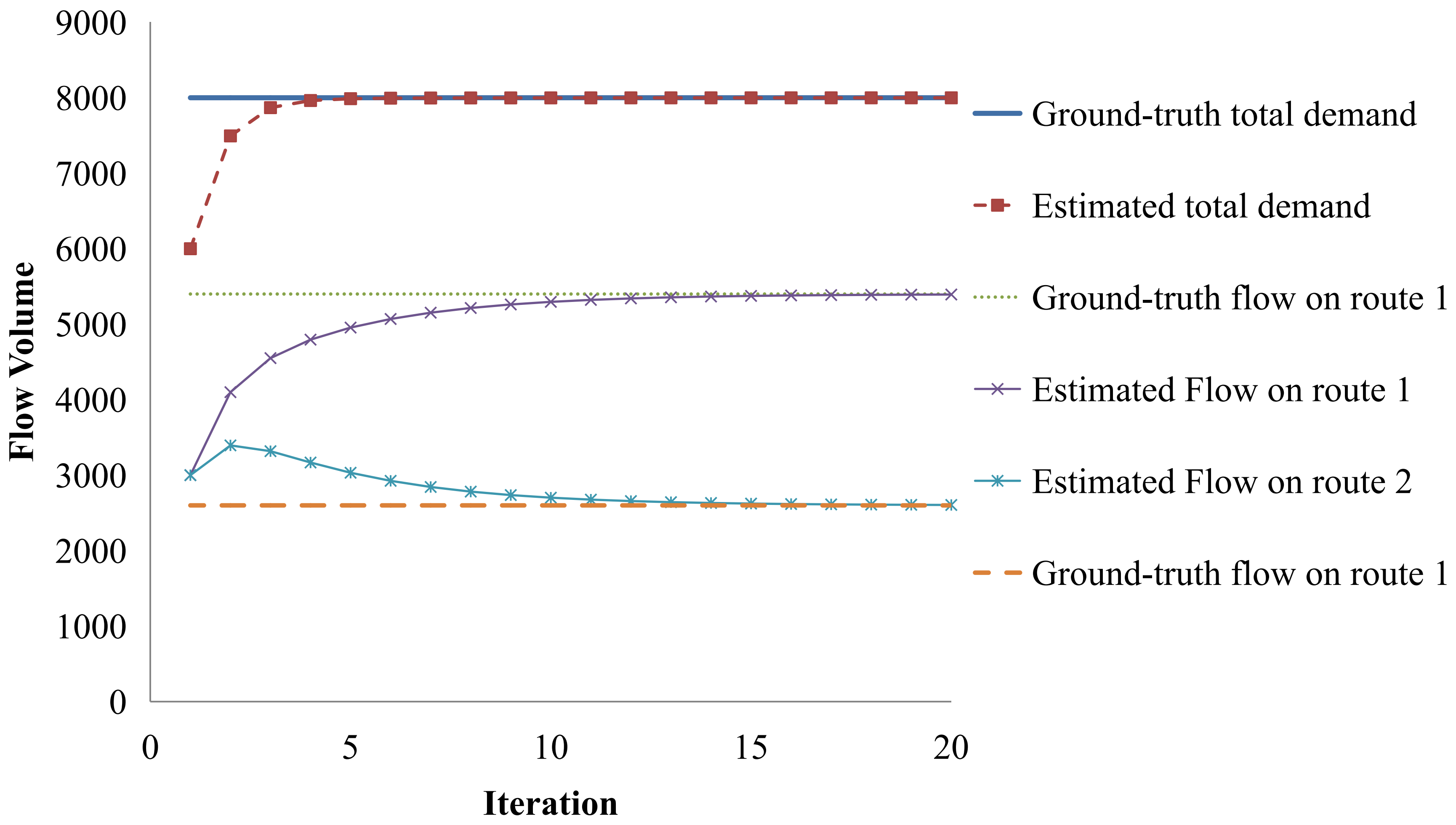
Congested two-link Corridor:

Total demand = 8000 veh/hour

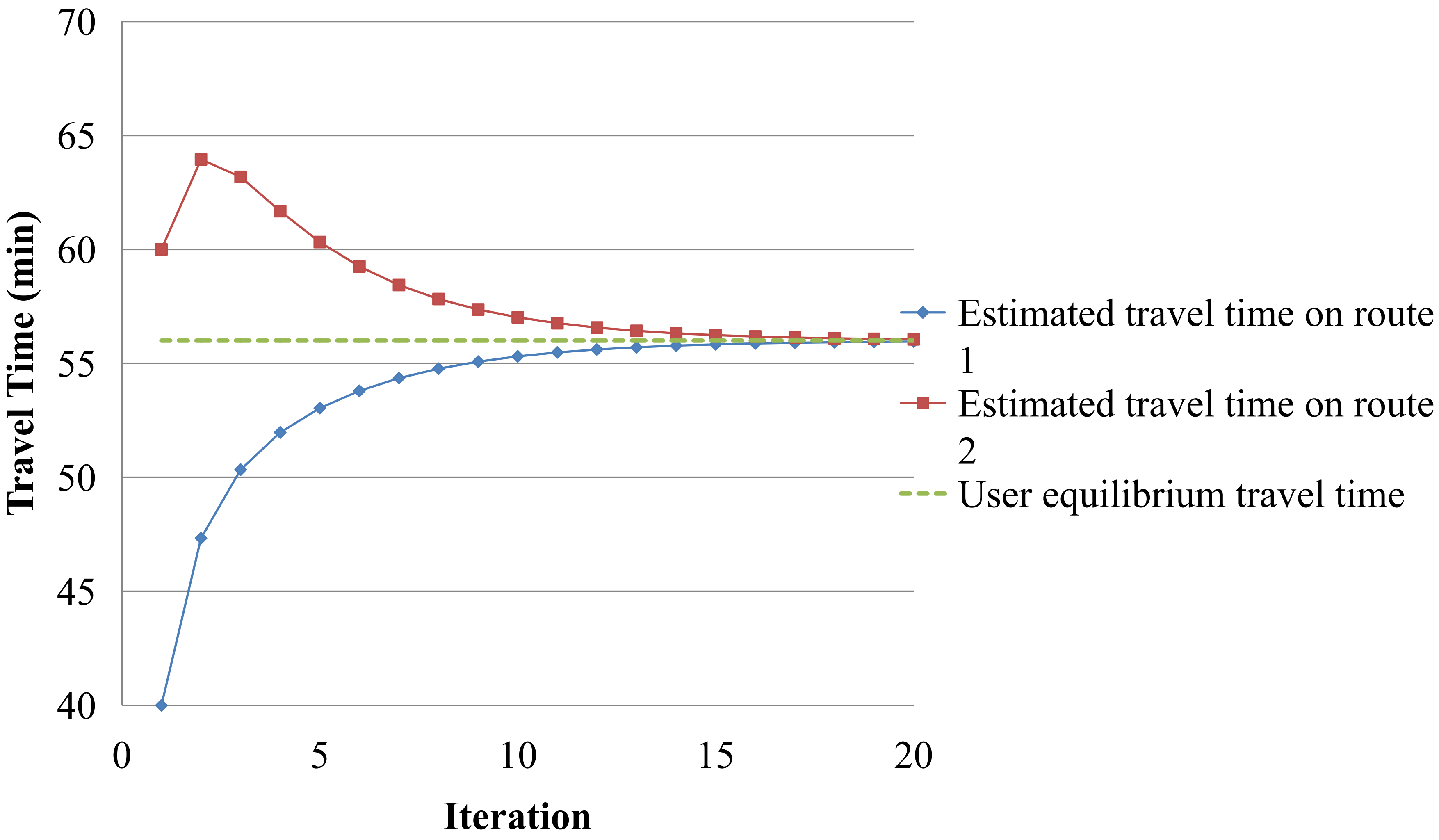



| Path | FFTT (min) | Capacity (veh/hr) | Assigned Flow (veh/hr) | Travel Time (min) |
|---|---|---|---|---|
| Path 1 | 20 | 3000 | 5400 | 56 |
| Path 2 | 30 | 3000 | 2600 | 56 |

# Path Flow Volume Convergence Pattern



# Path Travel Time Convergence Pattern

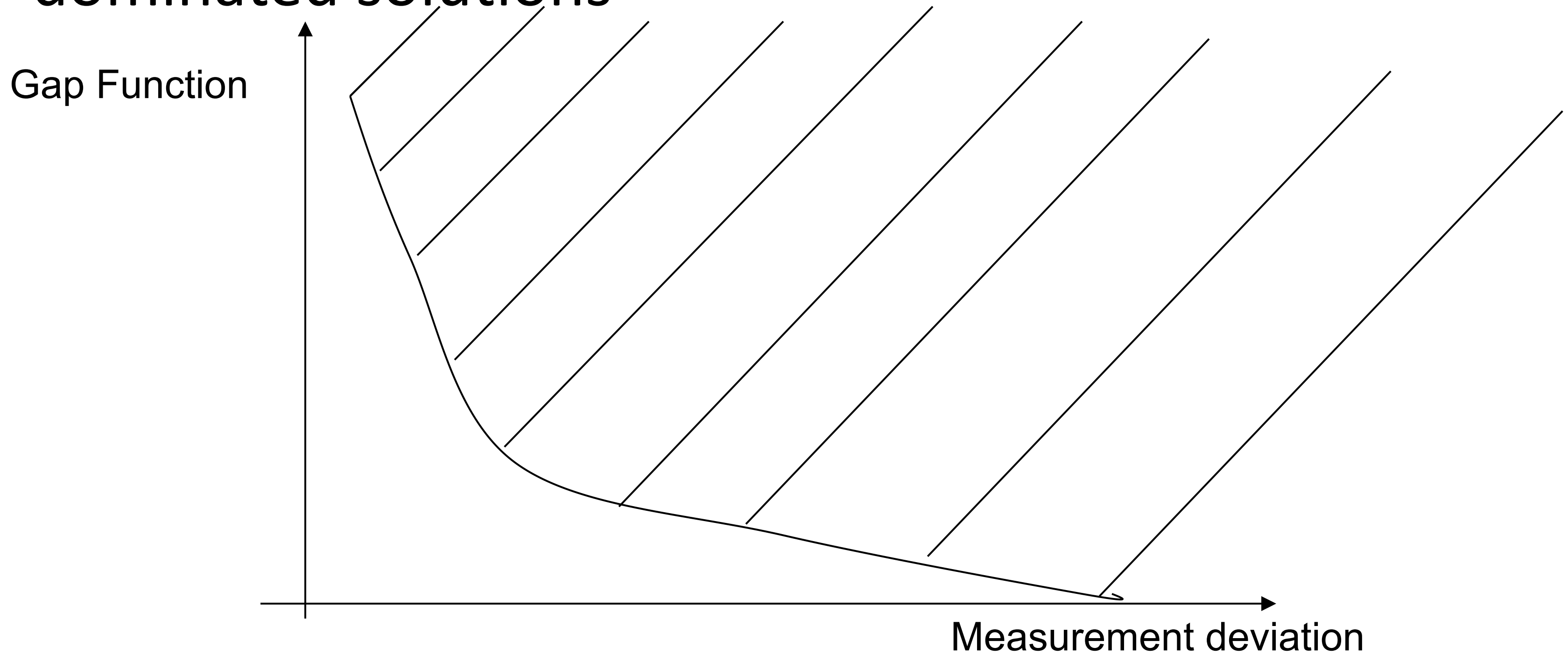


# Algorithm Performance under Different Input Conditions

| Information Availability | | | | Estimation Result | | | |
|---|---|---|---|---|---|---|---|
| Volume observations on path 1 only | Volume observations on path 2 only | Error-free target demand, 8000vhc/hr | Error-free travel time on path 1 | Flow on path 1 | Flow on path 2 | Total estimated demand | Equilibrium travel time (min) |
| X | | | | 5051.7 | 2367.8 | **7419.5** | **53.7** |
| | X | | | 4967.7 | 2311.8 | **7279.4** | **53.1** |
| X | X | | | 5011.8 | 2341.2 | **7353.0** | **53.4** |
| X | X | X | | 5387.9 | 2592.0 | **7979.9** | **55.9** |
| X | X | X | X | 5401.1 | 2600.7 | **8001.8** | **56.0** |

# Multi-objective Programming Perspective

- Gap Function: Degree of deviation from the original user equilibrium solutions
- Vary penalty on path cost to generate non-dominated solutions

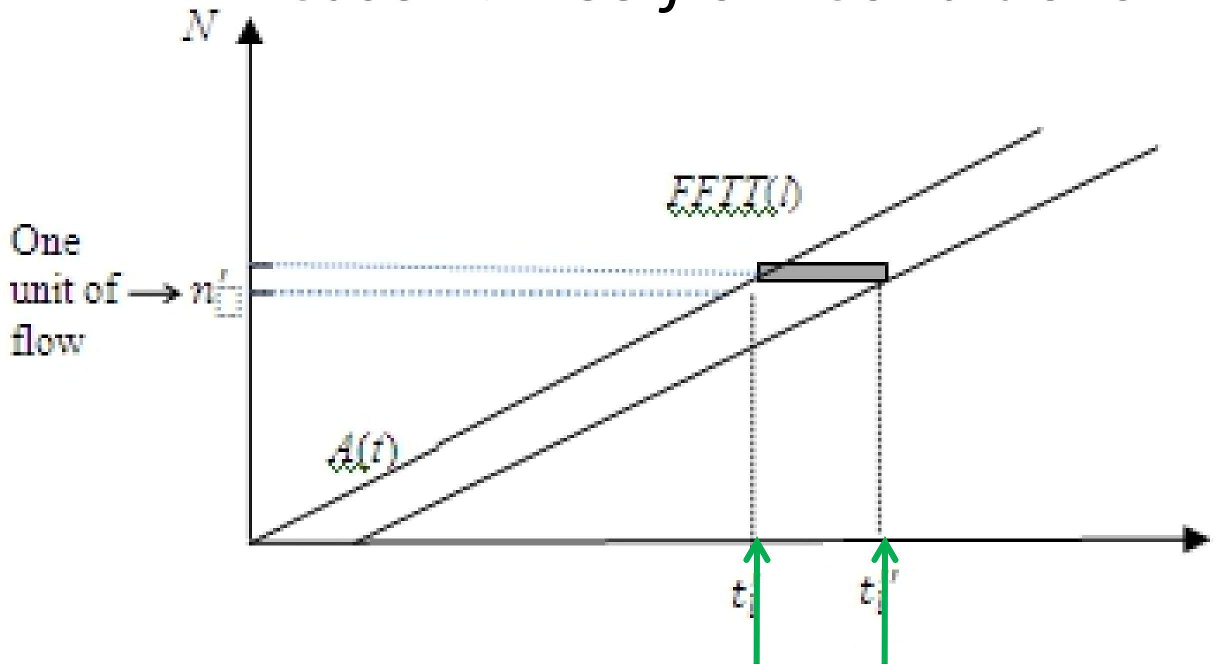


# Sensitivity-Analysis and Gradient Information

- Approximate the derivatives through simulation for each OD pair and each time interval in every iteration (Tavana, 2001)
- Gradient approximation methods
  - Simultaneous Perturbation Stochastic Approximation (SPSA) framework by Balakrishna et al. (2008); Cipriani et al. (2011)

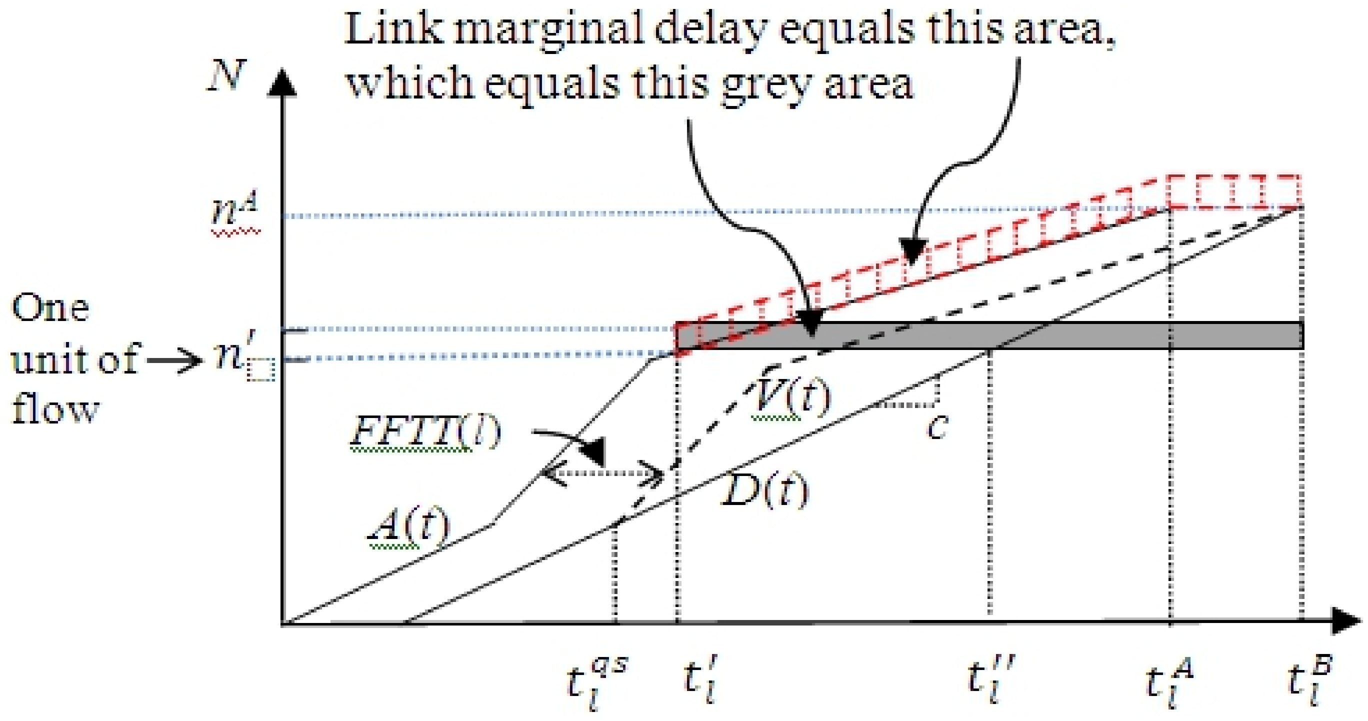
# How to Use Spatial Queue Model to Evaluate Partial Derivatives with respect to Path Flow Perturbation?

## Case 1: *Free-flow* Conditions



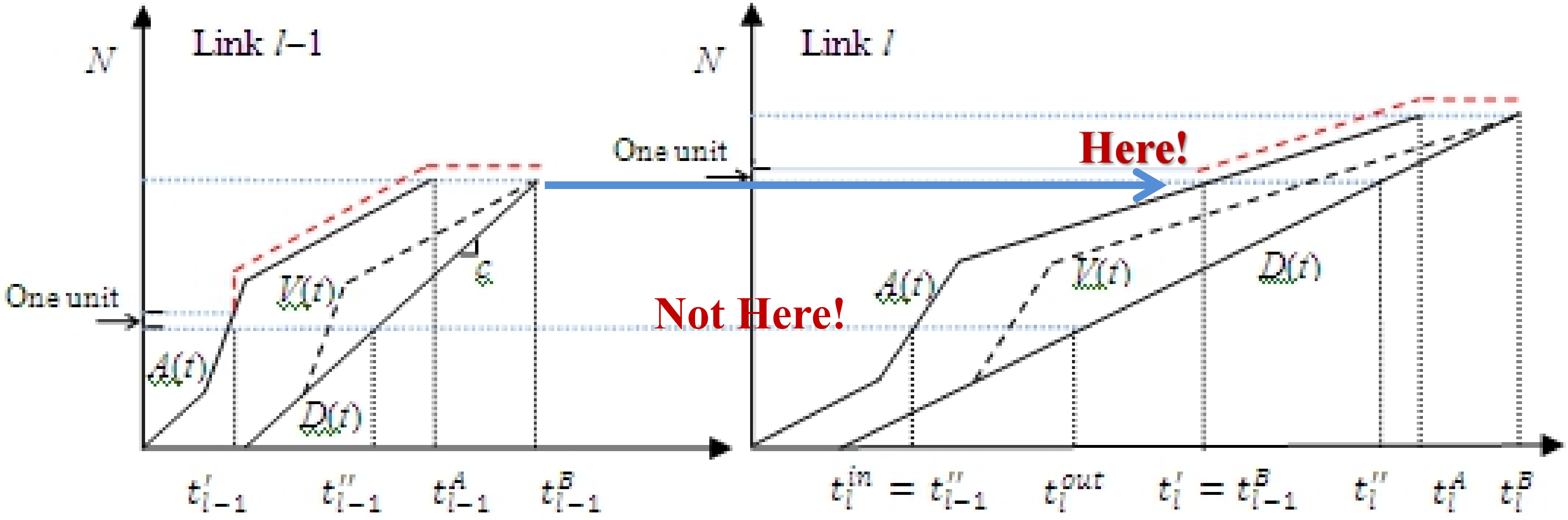

**Link inflow and outflow** increase by 1 at entering time and leaving time, respectively.

# Case 2: Congested Condition




Inspired by study by Ghali and Smith (1995)

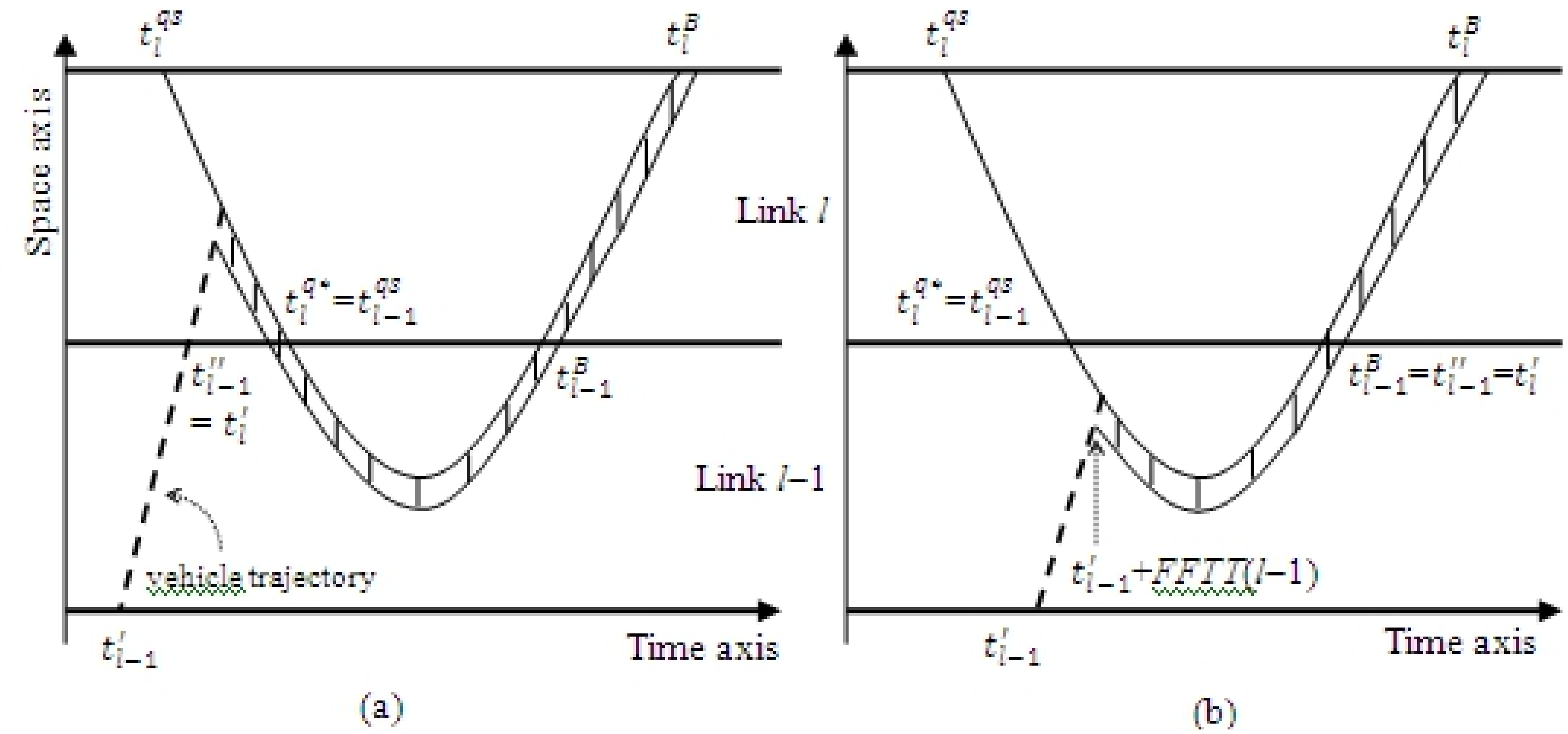
# Case 3: Two Partially Congested Links




The perturbation flow on the second link starts at the end of queue duration of the first link; rather than the vehicle entering time on second link

Similar work by Shen, Nie and Zhang (2007) for path marginal cost analysis

# Case 4: Queue Spillback




Individual extra delay depends on when the vehicle/perturbation flow joins in the queue.

# Summary

1. Integrated path-based OD demand estimation formulation

2. Gradient-based path flow adjustment process

3. Derive theoretically sound partial derivatives of link flow, density and travel time with respect to path flow perturbations

# Mathematical Details: Gradient Based Formulation

Adjust path flow on each path based on generalized gradient/Cost

$$
\begin{aligned}
r^{(m+1)}_{(w,\tau,p)} = \max \Big\{0, r^{(m)}_{(w,\tau,p)} \\
&- \gamma^{(m)} \Big[\beta_d \nabla f(r)\big|_{r=r^{(m)}} + \beta_q \nabla h^{q}(r)\big|_{r=r^{(m)}} + \beta_k \nabla h^{q}(r)\big|_{r=r^{(m)}} + \nabla h^{T}(r)\big|_{r=r^{(m)}} \\
&+ \lambda^{(n)} g(r,\pi)\big|_{r=r^{(m)}}\Big]\Big\}
\end{aligned}
$$

Individual gradients with respect to path flow adjustment

$$\nabla f(r) = \frac{\partial f(r)}{\partial r_{(w,\tau,p)}} = 2\Big(\sum_{\tau \in H_d} \sum_{p \in P} r_{(w,\tau,p)} - \bar{d}_{(w)}\Big)$$

$$\nabla h^{q}(r) = \frac{\partial h^{q}(r)}{\partial r_{(w,\tau,p)}} = 2\sum_{\tau \in H_o} \sum_{a \in S} \Big\{\big[q_{(a,t)}(r) - \bar{q}_{(a,t)}\big] \times \frac{\partial q_{(a,t)}(r)}{\partial r_{(w,\tau,p)}}\Big\}$$

$$\nabla h^{k}(r) = \frac{\partial h^{k}(r)}{\partial r_{(w,\tau,p)}} = 2\sum_{\tau \in H_o} \sum_{a \in S} \Big\{\big[k_{(a,t)}(r) - \bar{k}_{(a,t)}\big] \times \frac{\partial k_{(a,t)}(r)}{\partial r_{(w,\tau,p)}}\Big\}$$

$$\nabla h^{T}(r) = \frac{\partial h^{T}(r)}{\partial r_{(w,\tau,p)}} = 2\sum_{\tau \in H_o} \sum_{a \in S} \Big\{\big[T_{(a,t)}(r) - \bar{T}_{(a,t)}\big] \times \frac{\partial T_{(a,t)}(r)}{\partial r_{(w,\tau,p)}}\Big\}$$

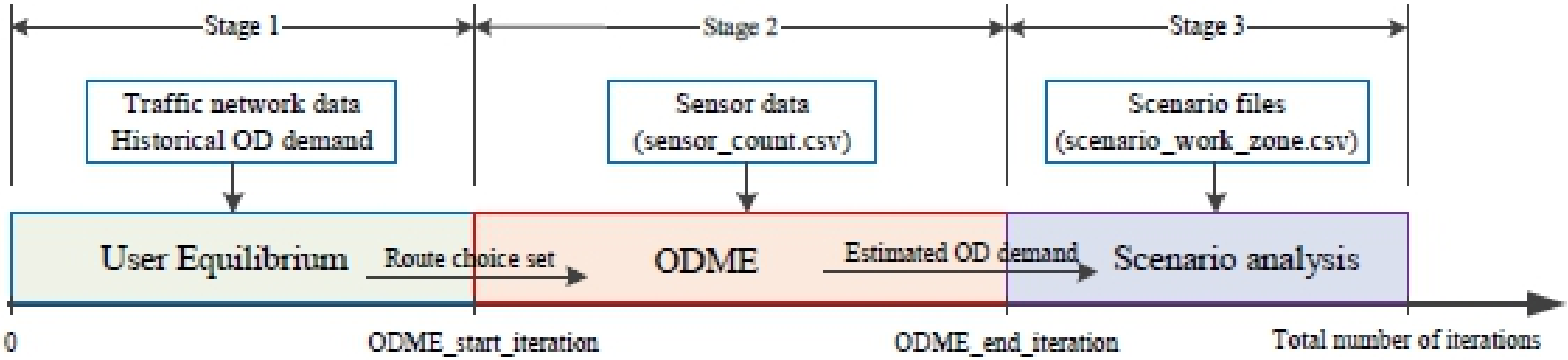
$$\nabla g(r,\pi) = \frac{\partial g(r,\pi)}{\partial r_{(w,\tau,p)}} = c_{(w,\tau,p)}(r) - \pi_{(w,\tau)} + r_{(w,\tau,p)} \frac{\partial c_{(w,\tau,p)}(r)}{\partial r_{(w,\tau,p)}}$$

Calculated based on the spatial queue model

# 6.1 OD Matrix Estimation Scenario

**The process of one simulation for scenario analysis using estimated OD demand**

- For performing scenario analysis, such as work zone, incident, ramp metering, etc., the travel demand should be calibrated in advance.
- In this section, the integration of the two parts above is realized in DTALite through just one simulation, the process of which can be illustrated in the following Figure



# 6.1 OD Matrix Estimation Scenario

## Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

- Input_link.csv file

- Set count sensor ID to “origin node -> destination node”
- Set speed sensor ID to “origin node -> destination node”

| B | C | D | E | F | G | O | P |
|---|---|---|---|---|---|---|---|
| link_id | from_node_id | to_node_id | direction | length | number_of_lanes | count_sensor_id | speed_sensor_id |
| 1285 | 1285 | 5018 | 1 | 0.2384 | 7 | 1285->5018 | 1285->5018 |
| 1286 | 1286 | 11125 | 1 | 0.466 | 7 | 1286->11125 | 1286->11125 |
| 1289 | 1289 | 4952 | 1 | 0.2427 | 7 | 1289->4952 | 1289->4952 |
| 1289 | 1289 | 5018 | 1 | 0.2621 | 7 | 1289->5018 | 1289->5018 |
| 1289 | 1289 | 11124 | 1 | 0.5069 | 7 | 1289->11124 | 1289->11124 |
| 1289 | 1289 | 11125 | 1 | 0.5047 | 7 | 1289->11125 | 1289->11125 |
| 1290 | 1290 | 4952 | 1 | 0.2516 | 7 | 1290->4952 | 1290->4952 |
| 1296 | 1296 | 5240 | 1 | 0.2056 | 7 | 1296->5240 | 1296->5240 |
| 1299 | 1299 | 4958 | 1 | 0.1988 | 7 | 1299->4958 | 1299->4958 |
| 1299 | 1299 | 11129 | 1 | 0.3637 | 7 | 1299->11129 | 1299->11129 |
| 1314 | 1314 | 11146 | 1 | 0.2758 | 7 | 1314->11146 | 1314->11146 |
| 1316 | 1316 | 11148 | 1 | 0.2608 | 7 | 1316->11148 | 1316->11148 |
| 1318 | 1318 | 11160 | 1 | 0.2384 | 7 | 1318->11160 | 1318->11160 |
| 1319 | 1319 | 11124 | 1 | 0.2054 | 7 | 1319->11124 | 1319->11124 |
| 1320 | 1320 | 5820 | 1 | 0.3445 | 7 | 1320->5820 | 1320->5820 |
| 1320 | 1320 | 11148 | 1 | 0.2527 | 7 | 1320->11148 | 1320->11148 |
| 1322 | 1322 | 5589 | 1 | 0.2229 | 7 | 1322->5589 | 1322->5589 |
| 1322 | 1322 | 11161 | 1 | 0.267 | 7 | 1322->11161 | 1322->11161 |
| 1323 | 1323 | 5592 | 1 | 0.2697 | 7 | 1323->5592 | 1323->5592 |

# 6.1 OD Matrix Estimation Scenario

## Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

- Input_scenario_setting.csv file
- so at least 30 iterations are required for the path flow adjustment for ODME
- Specify traffic assignment method to “3>ODME”

| E | F | G |
|---|---|---|
| traffic_flow_model | signal_representation_model | traffic_assignment_method |
| 1 | 0 | 3 |

# 6.1 OD Matrix Estimation Scenario

**Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario**

- Input_scenario_setting.csv file
- Specify start and end iteration of ODME (the iteration number indicate that ODME will begin at the 21th iteration and end at the 50th iteration)

| H | I | J | K |
|---|---|---|---|
| demand_multiplier | random_seed | ODME_start_iteration | ODME_end_iteration |
| 1 | 100 | 20 | 50 |

# 6.1 OD Matrix Estimation Scenario

## Step 2: Prepare Specific Input File for Signal Scenario

- Sensor_count.csv file
- Specify count sensor ID, origin node ID and destination node ID of observed links
- Specify observation time period of observed links

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| count_sensor_id | from_node_id | to_node_id | day_no | start_time_in_min | end_time_in_min | link_count |
| 5010->4958 | 5010 | 4958 | 1 | 990 | 1050 | 49.5 |
| 4958->5010 | 4958 | 5010 | 1 | 990 | 1050 | 74.5 |
| 4952->5022 | 4952 | 5022 | 1 | 990 | 1050 | 221.5 |
| 5022->4952 | 5022 | 4952 | 1 | 990 | 1050 | 147.5 |
| 5018->5436 | 5018 | 5436 | 1 | 990 | 1050 | 634 |
| 5436->5018 | 5436 | 5018 | 1 | 990 | 1050 | 951 |
| 4804->5109 | 4804 | 5109 | 1 | 990 | 1050 | 107 |
| 5109->4804 | 5109 | 4804 | 1 | 990 | 1050 | 160.5 |

# 6.1 OD Matrix Estimation Scenario

## Step 2: Prepare Specific Input File for Signal Scenario

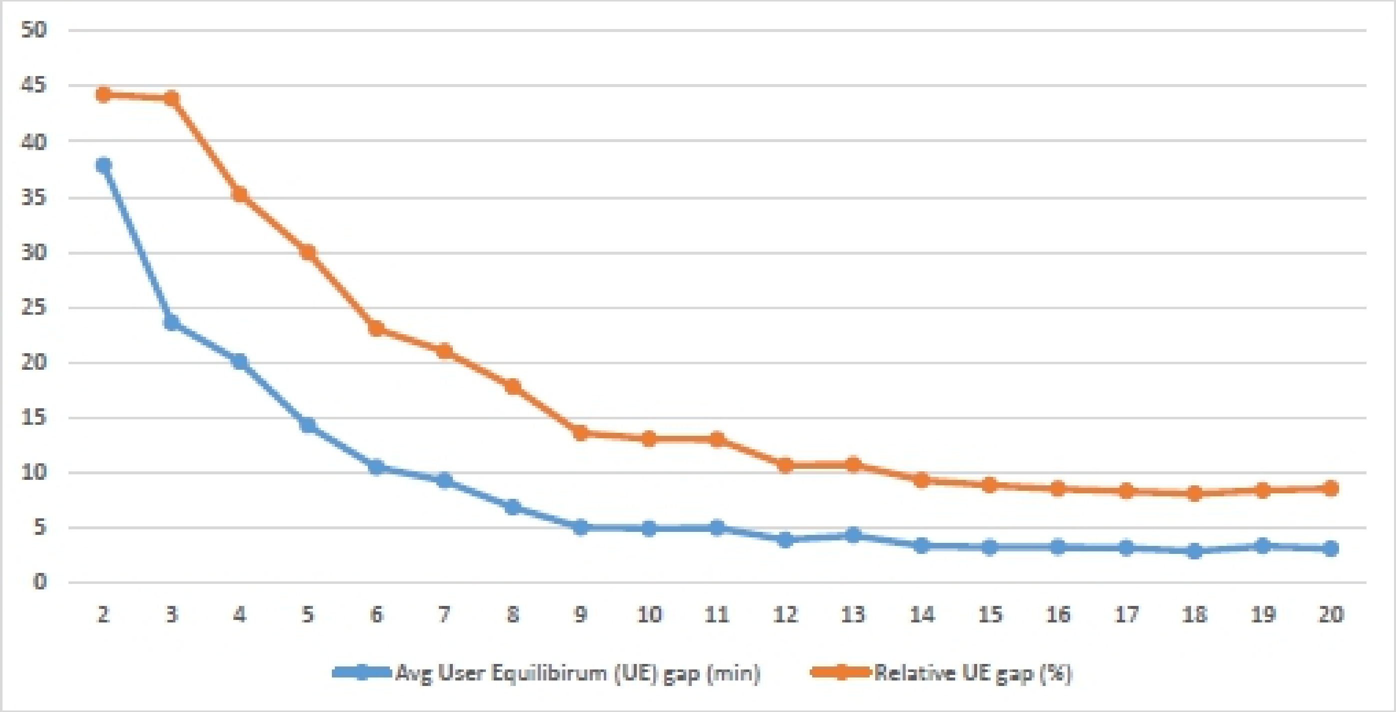
- Sensor_speed.csv file
- Specify count sensor ID, origin node ID and destination node ID of observed links
- Specify observation speed of observed links

| speed_sensor_id | from_node_id | to_node_id | day_no | start_time_in_min | end_time_in_min | speed |
|---|---|---|---|---|---|---|
| 5010->4958 | 5010 | 4958 | 1 | 990 | 1050 | 60 |
| 4958->5010 | 4958 | 5010 | 1 | 990 | 1050 | 60 |
| 4952->5022 | 4952 | 5022 | 1 | 990 | 1050 | 60 |
| 5022->4952 | 5022 | 4952 | 1 | 990 | 1050 | 60 |
| 5018->5436 | 5018 | 5436 | 1 | 990 | 1050 | 60 |
| 5436->5018 | 5436 | 5018 | 1 | 990 | 1050 | 60 |
| 4804->5109 | 4804 | 5109 | 1 | 990 | 1050 | 60 |
| 5109->4804 | 5109 | 4804 | 1 | 990 | 1050 | 60 |
| 6270->6273 | 6270 | 6273 | 1 | 990 | 1050 | 60 |
| 6273->6270 | 6273 | 6270 | 1 | 990 | 1050 | 60 |
| 5103->5820 | 5103 | 5820 | 1 | 990 | 1050 | 60 |
| 5820->5103 | 5820 | 5103 | 1 | 990 | 1050 | 60 |
| 5113->11148 | 5113 | 11148 | 1 | 990 | 1050 | 60 |
| 11148->5113 | 11148 | 5113 | 1 | 990 | 1050 | 60 |
| 6185->8172 | 6185 | 8172 | 1 | 990 | 1050 | 60 |

# 6.1 OD Matrix Estimation Scenario

## Step 3: Run Simulation and Perform Analysis

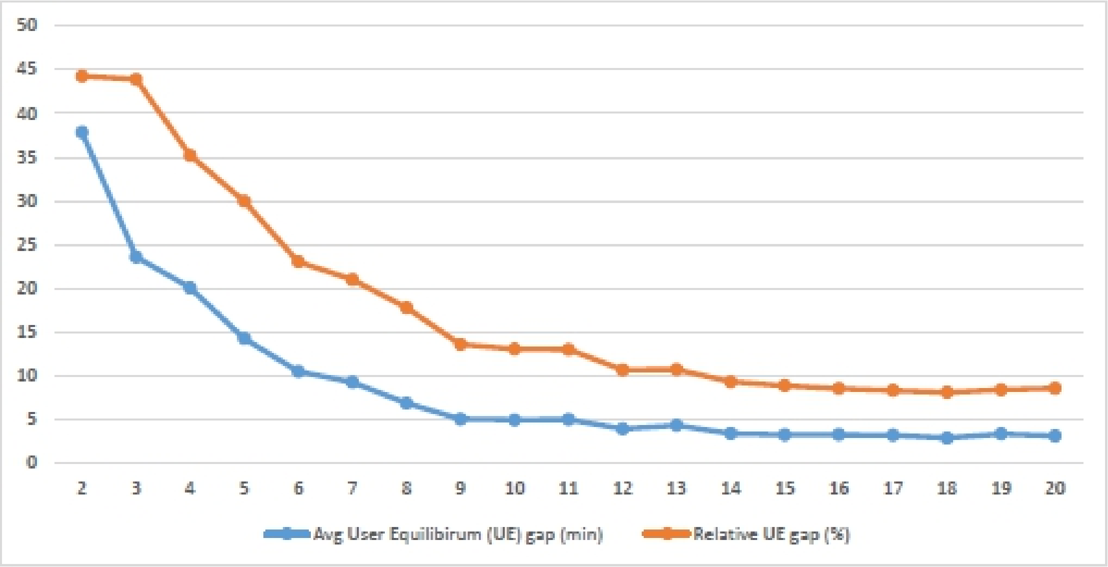
- checking output_summary.csv file, users can better understand the process of ODME in DTALite.



# 6.1 OD Matrix Estimation Scenario

## Step 3: Run Simulation and Perform Analysis

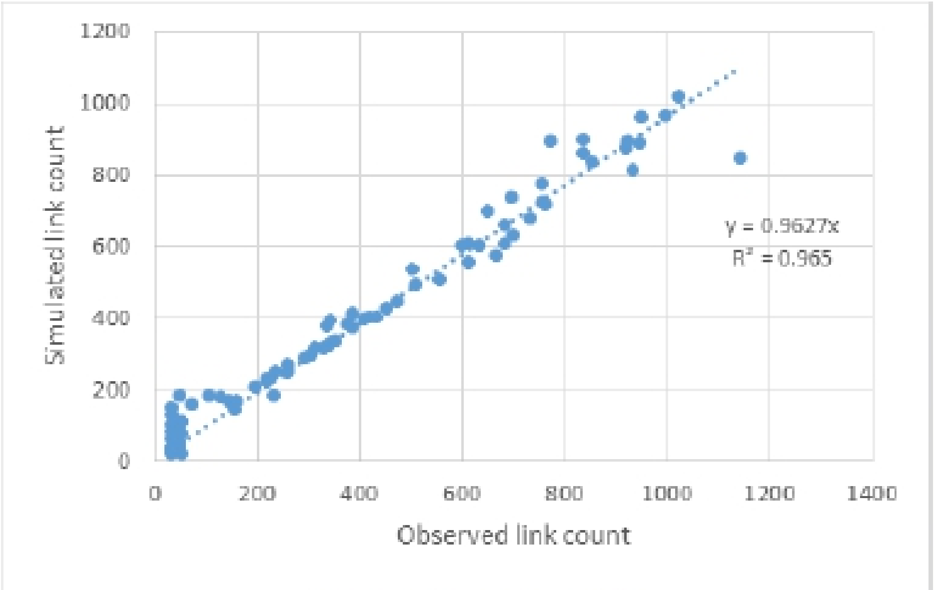
- For the first 20 iterations, a standard dynamic user equilibrium method, MAS, is used.
- It is expected to see the UE gap (Avg User Equilibirum (UE) gap (min) and Relative UE gap (%)) dramatically decreases and finally reach a stable state,



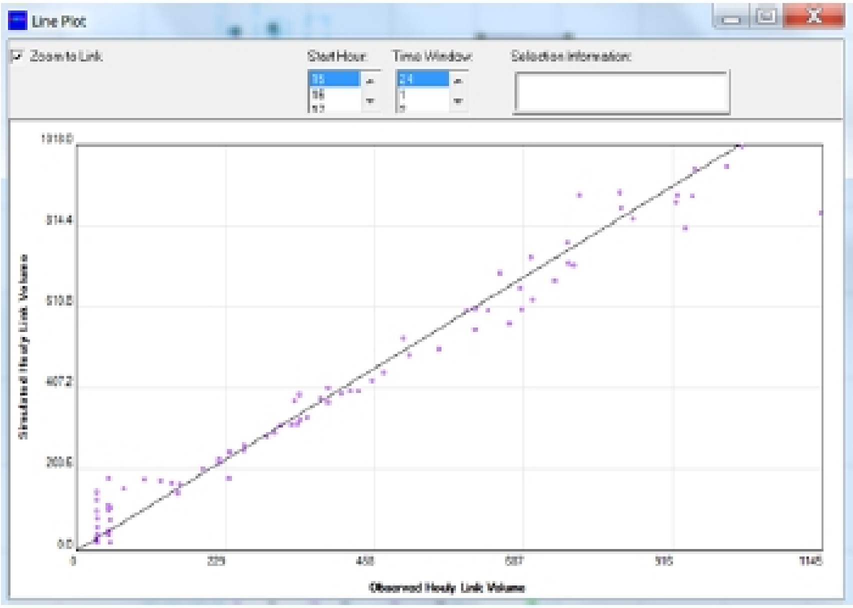
# 6.1 OD Matrix Estimation Scenario

## Step 3: Run Simulation and Perform Analysis

- Comparison result between observed link count and simulated link count after ODME




(a) comparison result in Excel



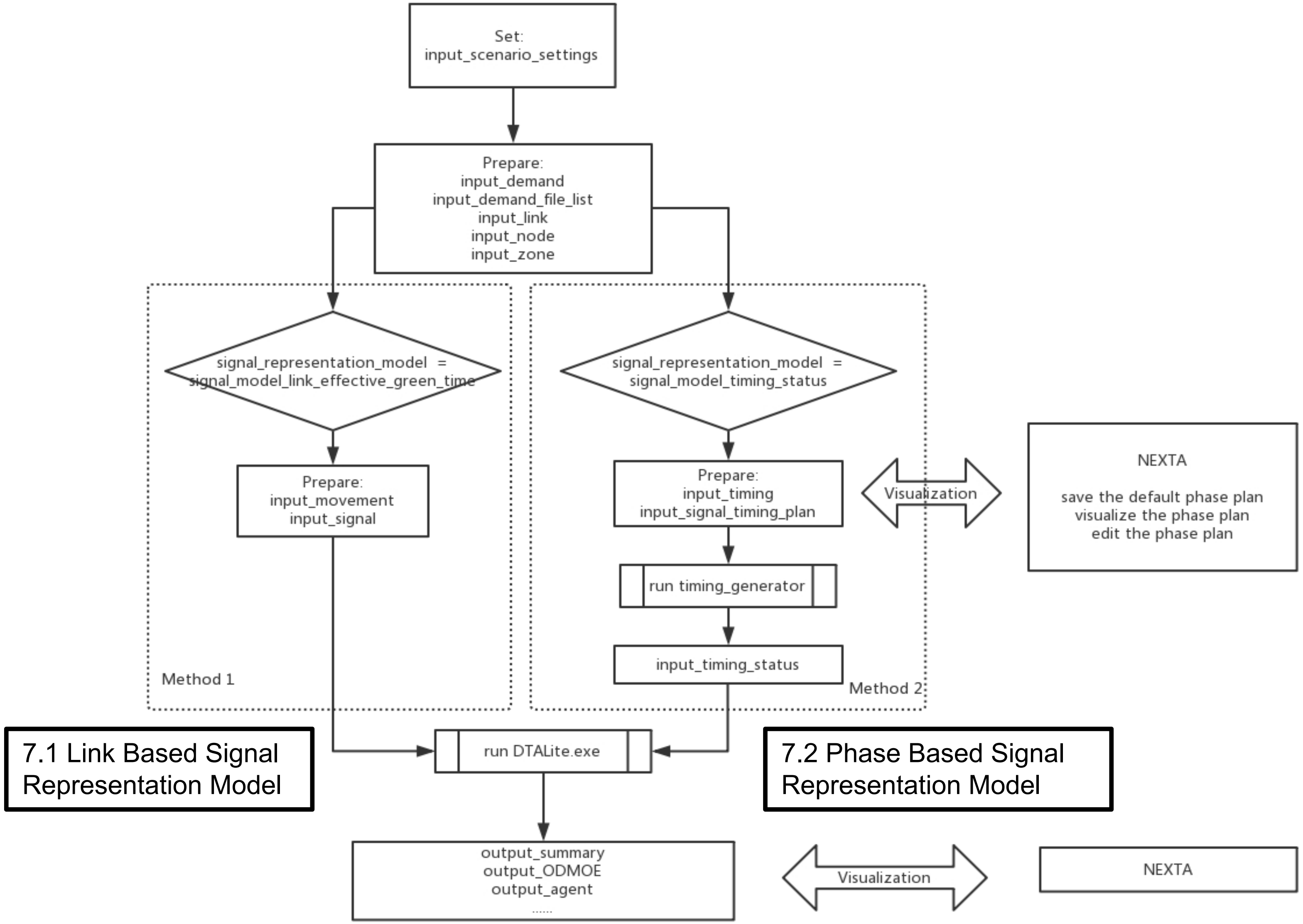

(b) comparison result in NeXTA

# Module 7
# Signalized Intersections Modeling

7.1 Link Based Signal Representation Model
7.2 Phase Based Signal Representation Model
(Contributed by Prof. Pengfei Taylor Li, Mississippi State University
http://www.cee.msstate.edu/pengfei-taylor-li/
Currently maintained by Jun Zhao from the University of Maryland Prof. Lei Zhang's Group )

7.3 Importing data from Synchro to DTALite

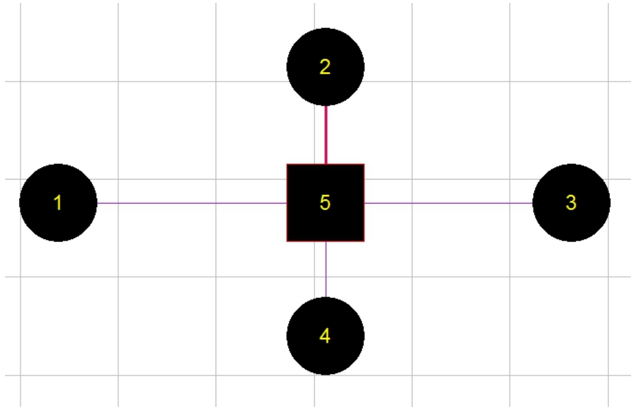
7.4. QEM (Signal Quick Estimation Method) Application to Generate signal timing data


Set:
input_scenario_settings
Prepare:
input_demand
input_demand_file_list
input_link
input_node
input_zone
signal_representation_model =
signal_model_link_effective_green_time
Prepare:
input_movement
input_signal
Method 1
signal_representation_model =
signal_model_timing_status
Prepare:
input_timing
input_signal_timing_plan
run timing_generator
input_timing_status
Method 2
Visualization
NEXTA
save the default phase plan
visualize the phase plan
edit the phase plan
7.1 Link Based Signal Representation Model
run DTALite.exe
7.2 Phase Based Signal Representation Model
output_summary
output_ODMOE
output_agent
......
Visualization
NEXTA

# 7.1 Link Based Signal Representation Model

## Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

- Input_node.csv
- Set the control type of signal intersections to signalized type
- (e.g., the control type of node 5 is set to “pretimed signal” in this case)

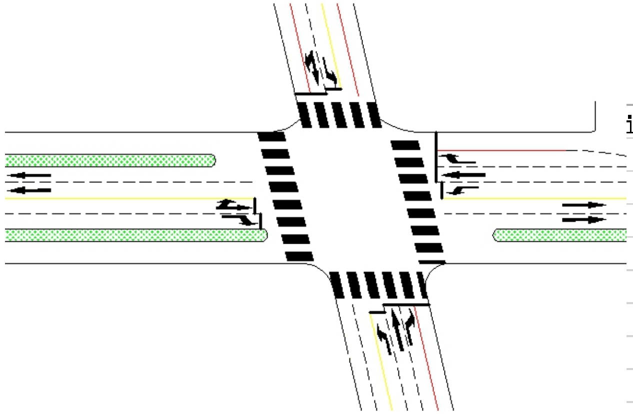



| B | C | D | E | F | G |
|---|---|---|---|---|---|
| node_id | control_type | control_type_name | x | y | geometry |
| 1 | 0 | unknown_control | 16.97894 | 23.79017 | <Point><coordinate |
| 5 | 5 | pretimed_signal | 30.58159 | 23.79017 | <Point><coordinate |
| 3 | 0 | unknown_control | 43.16913 | 23.79017 | <Point><coordinate |
| 2 | 0 | unknown_control | 30.58159 | 30.69301 | <Point><coordinate |
| 4 | 0 | unknown_control | 30.58159 | 16.98884 | <Point><coordinate |

# 7.1 Link Based Signal Representation Model

## Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

- Input_link.csv
- Specify the number of left-turn lanes (including separated left-turn lanes and shared left-turn lanes)



| B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|
| ink_id | from_node | to_node_i | directior | length | number_of_lanes | number_of_leftturn_lanes |
| 1 | 1 | 5 | 1 | 0.625 | 2 | 1 |
| 2 | 5 | 3 | 1 | 0.625 | 2 | |
| 3 | 2 | 5 | 1 | 0.625 | 2 | 1 |
| 4 | 5 | 2 | 1 | 0.625 | 1 | |
| 5 | 5 | 4 | 1 | 0.625 | 1 | |
| 6 | 4 | 5 | 1 | 0.625 | 3 | 1 |
| 7 | 3 | 5 | 1 | 0.625 | 3 | 1 |
| 8 | 5 | 1 | 1 | 0.625 | 2 | |

# 7.1 Link Based Signal Representation Model

## Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

- Input_scenario_setting.csv
- Set traffic flow model to “2 -> Spatial Queue Model”
- Set signal representation model to “1 -> link based signal

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| scenario | scenari | number_of_iterations | traffic_flow_model | signal_representation_model | traffic_analysis_method |
| 1 | test1 | 1 | 2 | 1 | 0 |

# 7.1 Link Based Signal Representation Model

## Step 2: Prepare Specific Input File for Signal Scenario

- Input_movement.csv
- Specify left-turn type movements and through-type movements which are required

| node_id | up_node_id | dest_node_id | name | turn_type | prohibited_flag |
|---|---|---|---|---|---|
| 5 | 1 | 2 | | Left | 0 |
| 5 | 2 | 3 | | Left | 0 |
| 5 | 3 | 4 | | Left | 0 |
| 5 | 4 | 1 | | Left | 0 |
| 5 | 1 | 3 | | Through | 0 |
| 5 | 2 | 4 | | Through | 0 |
| 5 | 3 | 1 | | Through | 0 |
| 5 | 4 | 2 | | Through | 0 |
| 5 | 1 | 4 | | Right | 0 |
| 5 | 2 | 1 | | Right | 0 |
| 5 | 3 | 2 | | Right | 0 |
| 5 | 4 | 3 | | Right | 0 |

# 7.1 Link Based Signal Representation Model

## Step 2: Prepare Specific Input File for Signal Scenario

- Input_signal.csv
- the number of phase plan
- starting time(in seconds), ending time(in seconds), cycle length(in seconds) and offset (in seconds) of each signal plan

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| from_node | to_node | plan_no | plan_startime_in_sec | plan_endtime_in_sec | to_node_cycle_in_sec | to_node_offset_in_sec |
| 1 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 2 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 3 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 4 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 1 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 2 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 3 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 4 | 5 | 2 | 1200 | 864000 | 120 | 0 |

# 7.1 Link Based Signal Representation Model

## Step 2: Prepare Specific Input File for Signal Scenario

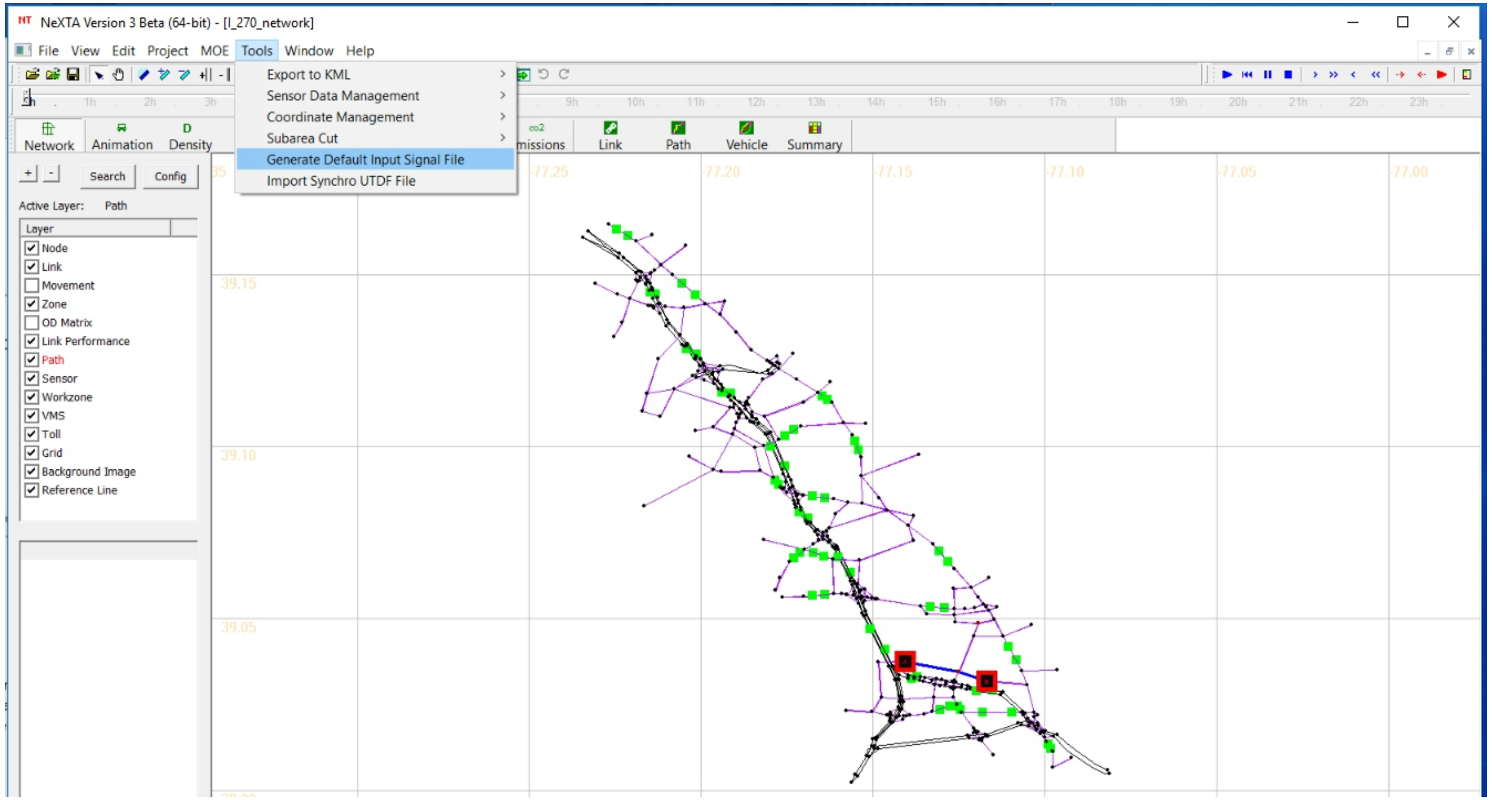
- Input_signal.csv
- saturation flow(per hour per lane), start time(in seconds) and end time(in seconds) for through-type movement and left-turned movement

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| from_node | to_node | plan_no | plan_startime_in_sec | plan_endtime_in_sec | to_node_cycle_in_sec | to_node_offset_in_sec |
| 1 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 2 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 3 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 4 | 5 | 1 | 1 | 1200 | 190 | 0 |
| 1 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 2 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 3 | 5 | 2 | 1200 | 864000 | 120 | 0 |
| 4 | 5 | 2 | 1200 | 864000 | 120 | 0 |

## Step 3: Run Simulation

# 7.1 Link Based Signal Representation Model

## PS: Generate Default Input Signal Files




- For the intersections without signal timing, NEXTA could generate the default signal files in input_signal_default.csv.

# 7.2 Phase Based Signal Representation Model

**Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario**

- Input_node.csv
- Input_link.csv
- Setup data in these two files the same as “7.1 Link Based Signal Representation Model”

# 7.2 Phase Based Signal Representation Model

## Step 2: Parameter settings in key simulation configuration files

- Input_movement.csv
- Movement id
- turn direction

| movement_id | node_id | up_node_id | dest_node_id | name | turn_type | turn_direction | prohibited_flag |
|---|---|---|---|---|---|---|---|
| 1-5-2-L | 5 | 1 | 2 | WBL | Left | WBL | 0 |
| 1-5-3-T | 5 | 1 | 3 | WBT | Through | WBT | 0 |
| 1-5-4-R | 5 | 1 | 4 | WBR | Right | WBR | 0 |
| 2-5-3-L | 5 | 2 | 3 | NBL | Left | NBL | 0 |
| 2-5-4-T | 5 | 2 | 4 | NBT | Through | NBT | 0 |
| 2-5-1-R | 5 | 2 | 1 | NBR | Right | NBR | 0 |
| 3-5-4-L | 5 | 3 | 4 | EBL | Left | EBL | 0 |
| 3-5-1-T | 5 | 3 | 1 | EBT | Through | EBT | 0 |
| 3-5-2-R | 5 | 3 | 2 | EBR | Right | EBR | 0 |
| 4-5-1-L | 5 | 4 | 1 | SBL | Left | SBL | 0 |
| 4-5-2-T | 5 | 4 | 2 | SBT | Through | SBT | 0 |
| 4-5-3-R | 5 | 4 | 3 | SBR | Right | SBR | 0 |

# 7.2 Phase Based Signal Representation Model

## Step 2: Parameter settings in key simulation configuration files

- input_signal_timing_plan.csv
- Intersection id
- Timing plan number
- Start time and end time of timing plan
- Starting phase
- Offset

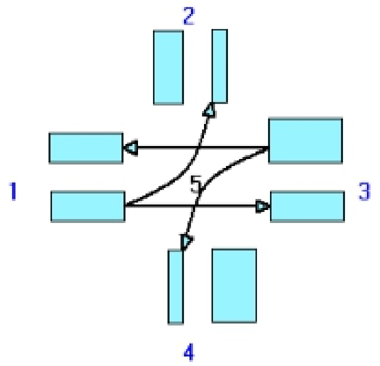
| [TOD] | int_id | timing_plan_no | start_time_in_sec | end_time_in_sec | starting_phase | off_set |
|---|---|---|---|---|---|---|
| | 5 | 1 | 0 | 86400 | 1 | 11 |

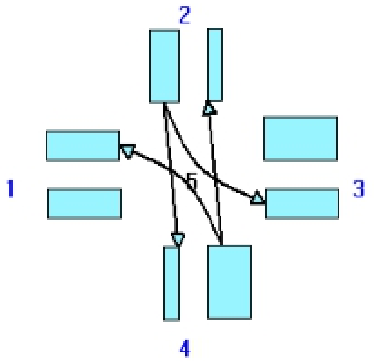
- This shows that the intersection 5, has one timing plan from time 0 to 86400 seconds. The starting phase is 1 and offset is 11 seconds.

# 7.2 Phase Based Signal Representation Model

## Step 2: Parameter settings in key simulation configuration files

- input_timing.csv
- Intersection id
- Timing plan number
- Number of phase
- green duration, movement string and movement direction string of phase




| [Signal] | int_id | timing_plan_no | phase_id | next_phase | green_duration | movement_str | movement_dir_str |
|---|---|---|---|---|---|---|---|
| | 5 | 1 | 1 | 2 | 120 | 1_3_T; 1_2_L; 3_1_T; 3_4_L | WBT; WBL; EBT; EBL; |
| | 5 | 1 | 2 | 1 | 70 | 2_4_T; 2_3_L; 4_2_T; 4_1_L | SBT; SBL; NBT; NBL; |

# 7.2 Phase Based Signal Representation Model

**Step 3:** generate the input_timing_status.csv file with timing_generator.exe

- Required files include input_timing.csv file and input_signal_timing_plan.csv file

| movement_str | start_time_in_sec | end_time_in_sec | signal_status | from_node_id | to_node_id | turn_type |
|---|---|---|---|---|---|---|
| **1_3_T** | 0 | 119 | 1 | 1 | 5 | T |
| **1_2_L** | 0 | 119 | 1 | 1 | 5 | L |
| **3_1_T** | 0 | 119 | 1 | 3 | 5 | T |
| **3_4_L** | 0 | 119 | 1 | 3 | 5 | L |
| **…** | … | … | … | … | … | … |

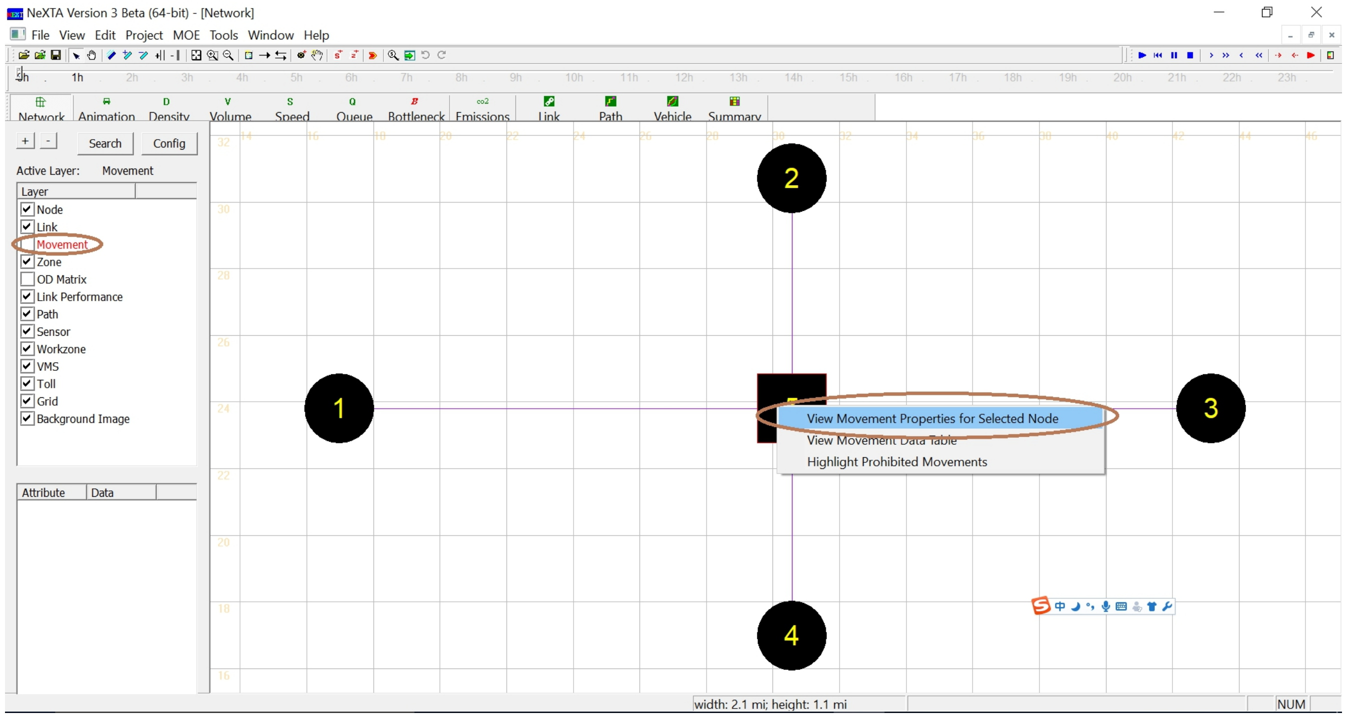
C:\DTALite\DTALite_related_from_DRZhou_0804\Shared signal test data_0820\0-Signal_tes...

```
the int_id is: 5
===========================================
Intersection: 5
plan_id: 1
Press any key to continue . . .
```

# 7.2 Phase Based Signal Representation Model

## Step 3: NEXTA GUI instruction

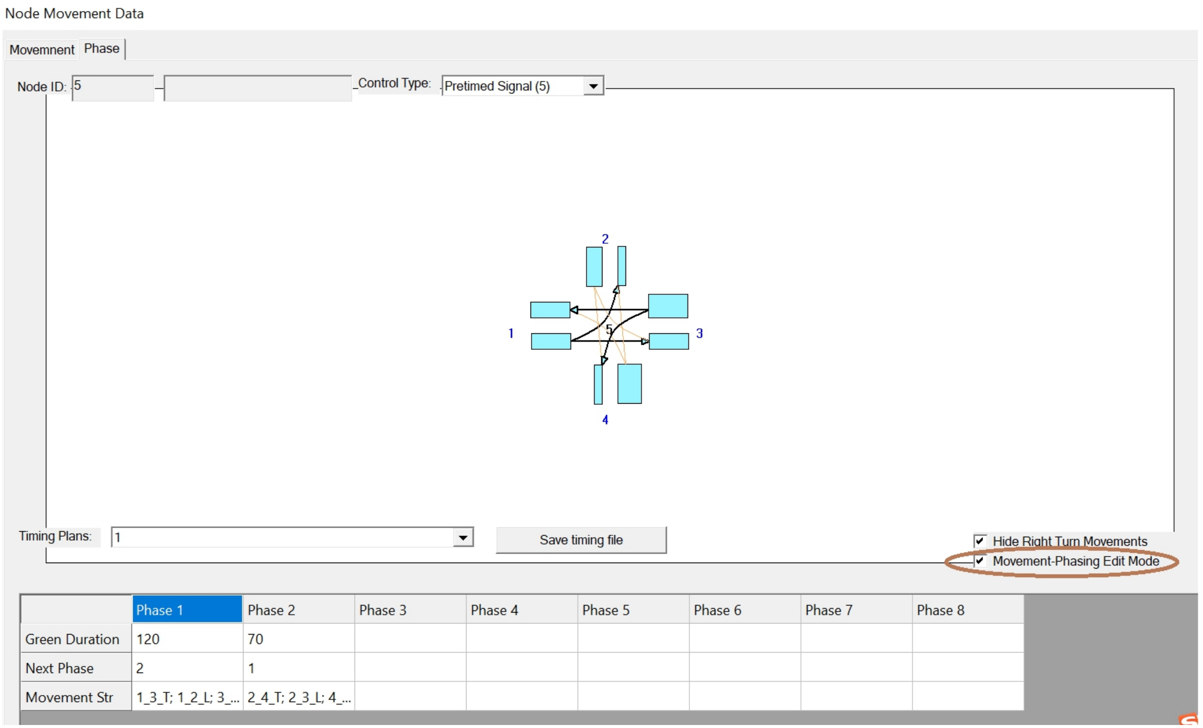
- Load the example project
- Choose the Movement layer -> right click the intersection node -> View Movement Properties for Selected Node.



# 7.2 Phase Based Signal Representation Model

## Step 3: NEXTA GUI instruction

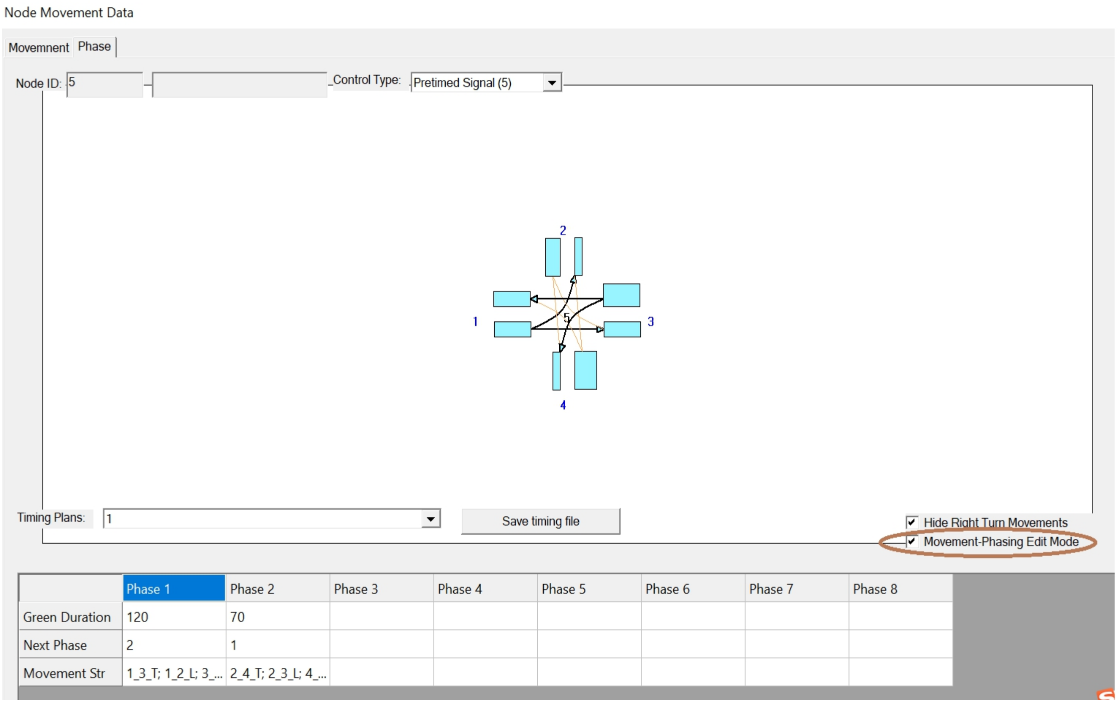
- If this section has the phase plan before, thus we could display them in NEXTA.



# 7.2 Phase Based Signal Representation Model

## Step 4: Add/delete a phase in current phase

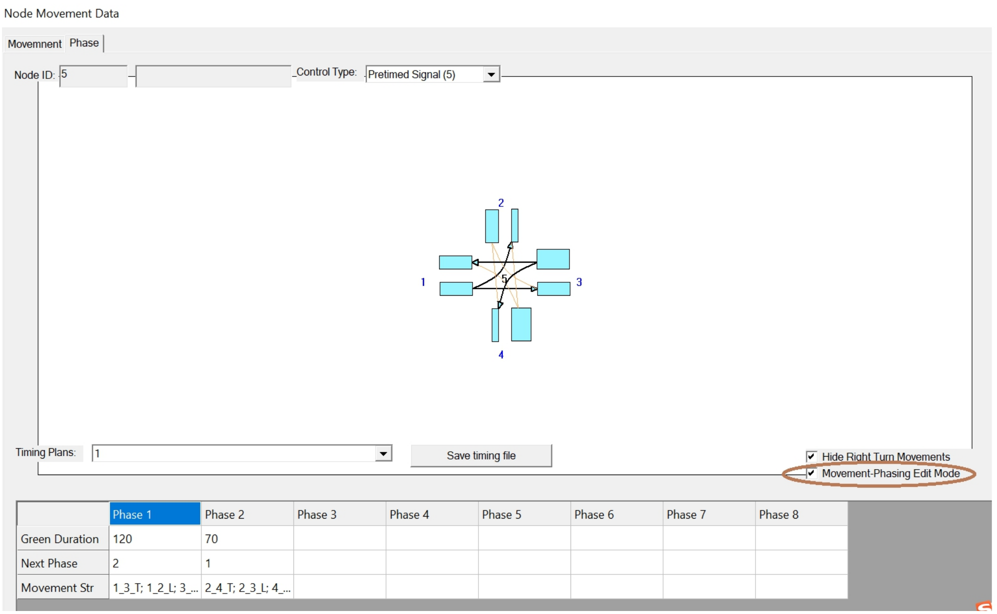
- Add/delete a phase in current phase:
- Check the box before Movement-Phasing Edit Mode
- Double click the phase that you want to add/delete



# 7.2 Phase Based Signal Representation Model

## Step 4: Add/delete a phase in current phase

- then the corresponding movement_str will be changed in the table below
- Click save timing file button, the input_timing file will be updated in the folder.



# 7.2 Phase Based Signal Representation Model

## Step 5: Run Simulation and Perform Analysis

- Input_scenario_setting.csv
- Set traffic flow model to “2 -> Spatial Queue Model”
- Set signal representation model to “2 -> phase based signal”

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| scenario_ | scenari | number_of_iterations | traffic_flow_model | signal_representation_model | traffic_analysis_method |
| 1 | test1 | 1 | 2 | 2 | 0 |

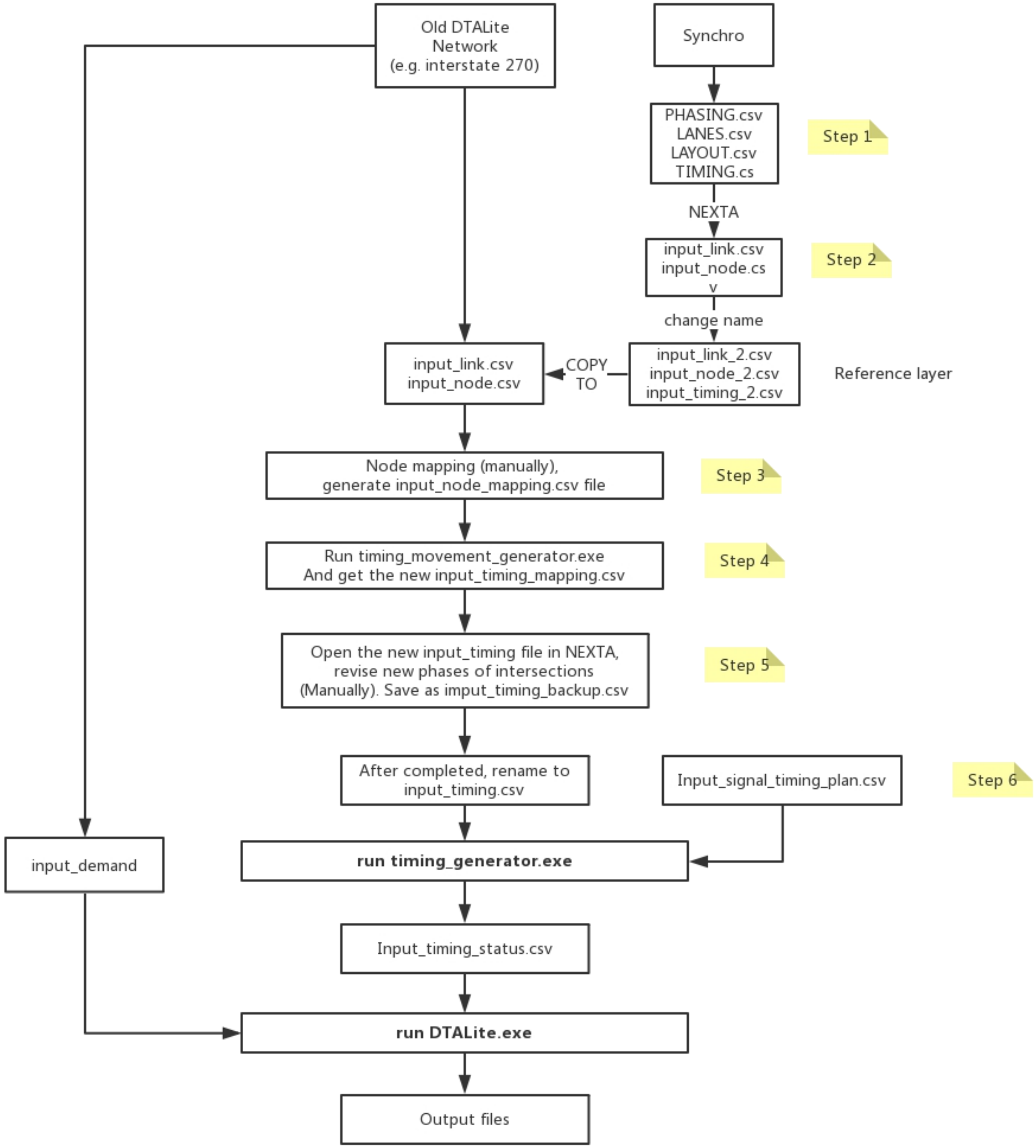
# 7.2 Phase Based Signal Representation Model

## Step 5: Run Simulation and Perform Analysis

- The results look like this:

| agent_id | from_origi | to_destinati | departure_time_ | arrival_time_ | comp | travel_time_in_min |
|---|---|---|---|---|---|---|
| 6 | 2 | 4 | 0.59 | 3.6 | c | 3.01 |
| 23 | 2 | 4 | 1.19 | 4.2 | c | 3.01 |
| 26 | 2 | 4 | 1.78 | 4.8 | c | 3.02 |
| 45 | 2 | 4 | 2.38 | 6.8 | c | 4.42 |
| 52 | 2 | 4 | 2.97 | 6.9 | c | 3.93 |
| 64 | 2 | 4 | 3.56 | 6.9 | c | 3.34 |
| 74 | 2 | 4 | 4.16 | 7.2 | c | 3.04 |
| 90 | 2 | 4 | 4.75 | 7.8 | c | 3.05 |
| 106 | 2 | 4 | 5.35 | 10 | c | 4.65 |
| 118 | 2 | 4 | 5.94 | 10.1 | c | 4.16 |
| 129 | 2 | 4 | 6.53 | 10.1 | c | 3.57 |
| 142 | 2 | 4 | 7.13 | 10.2 | c | 3.07 |
| 152 | 2 | 4 | 7.72 | 10.8 | c | 3.08 |

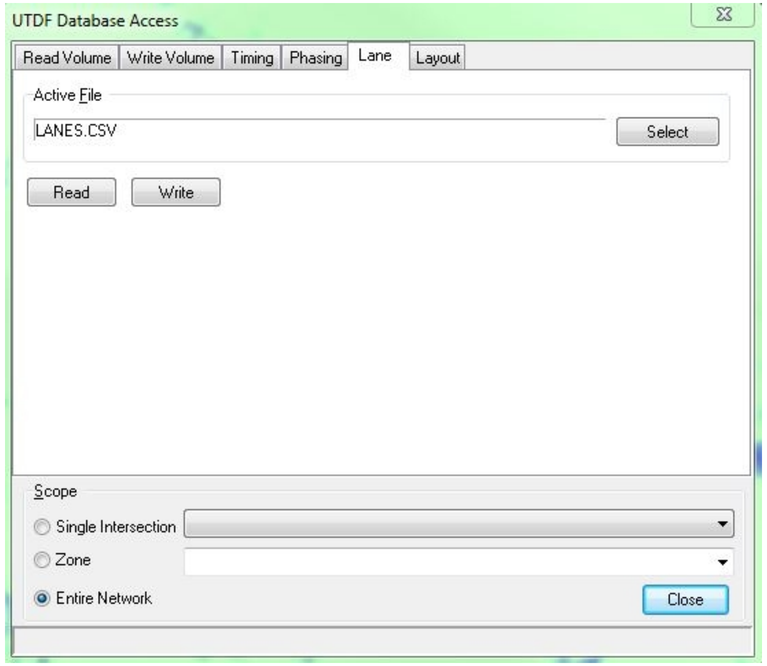
# 7.3 Importing data from synchro to DTALite

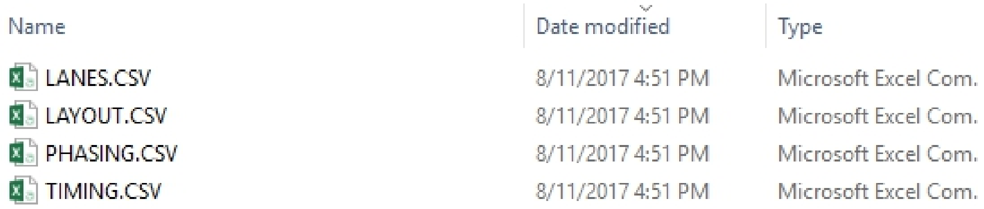


# 7.3 Importing data from synchro to DTALite

## Step 1: Prepare the Synchro dataset

- Export to LANES.cvs, LAYOUT.csv, PHASING.csv and TIMING.CSV (UTDF format).

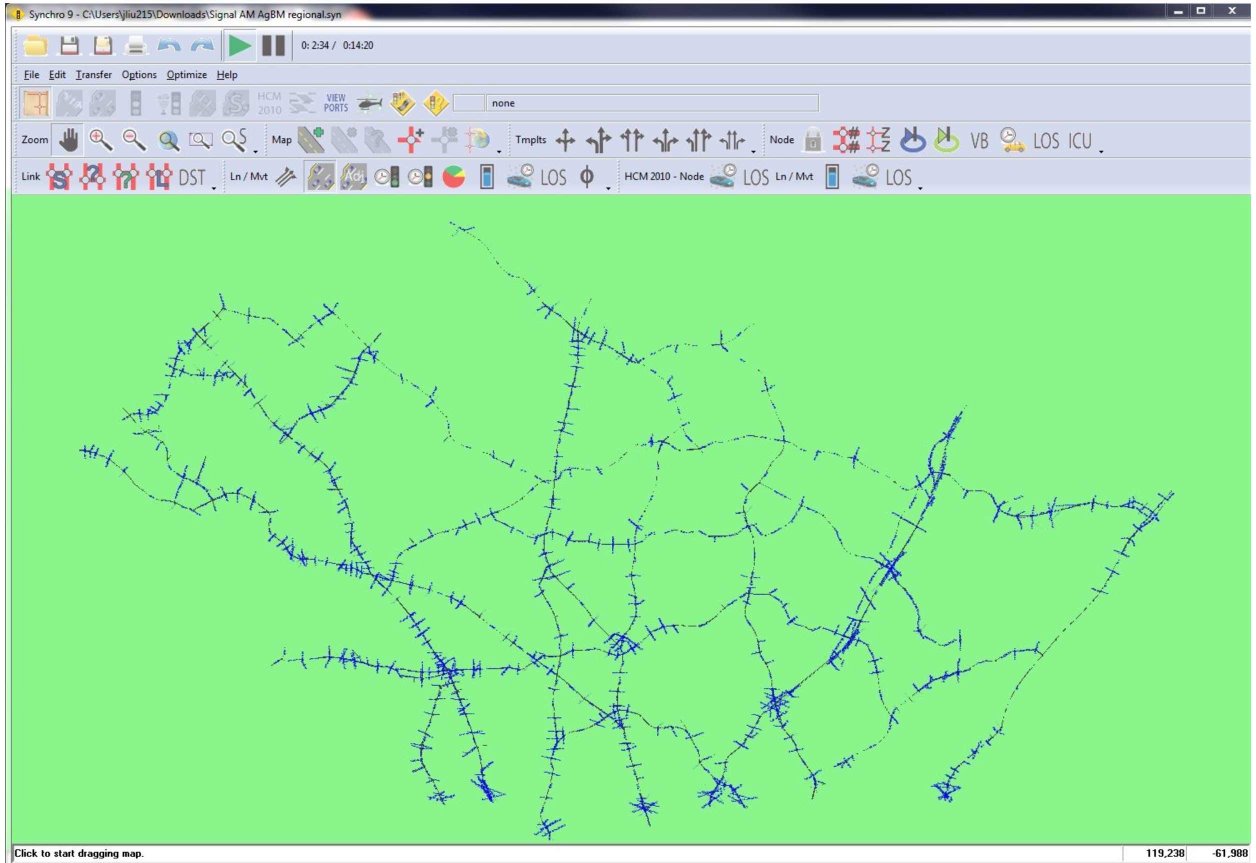


# 7.3 Importing data from synchro to DTALite

## Step 1: Prepare the Synchro dataset

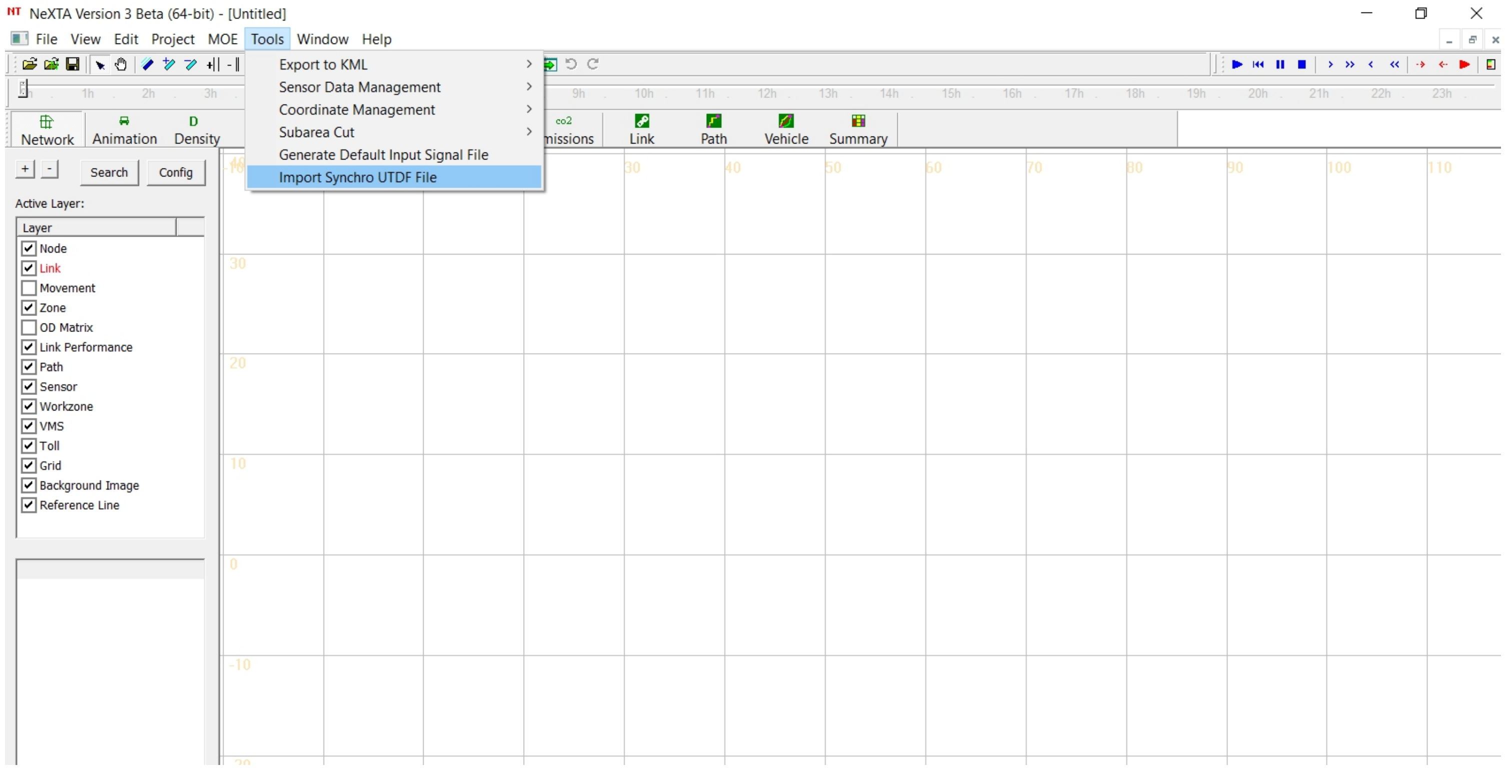
- Export to LANES.cvs, LAYOUT.csv, PHASING.csv and TIMING.CSV (UTDF format).



# 7.3 Importing data from synchro to DTALite

## Step 2: Use nexta to import the synchro UTDF format.

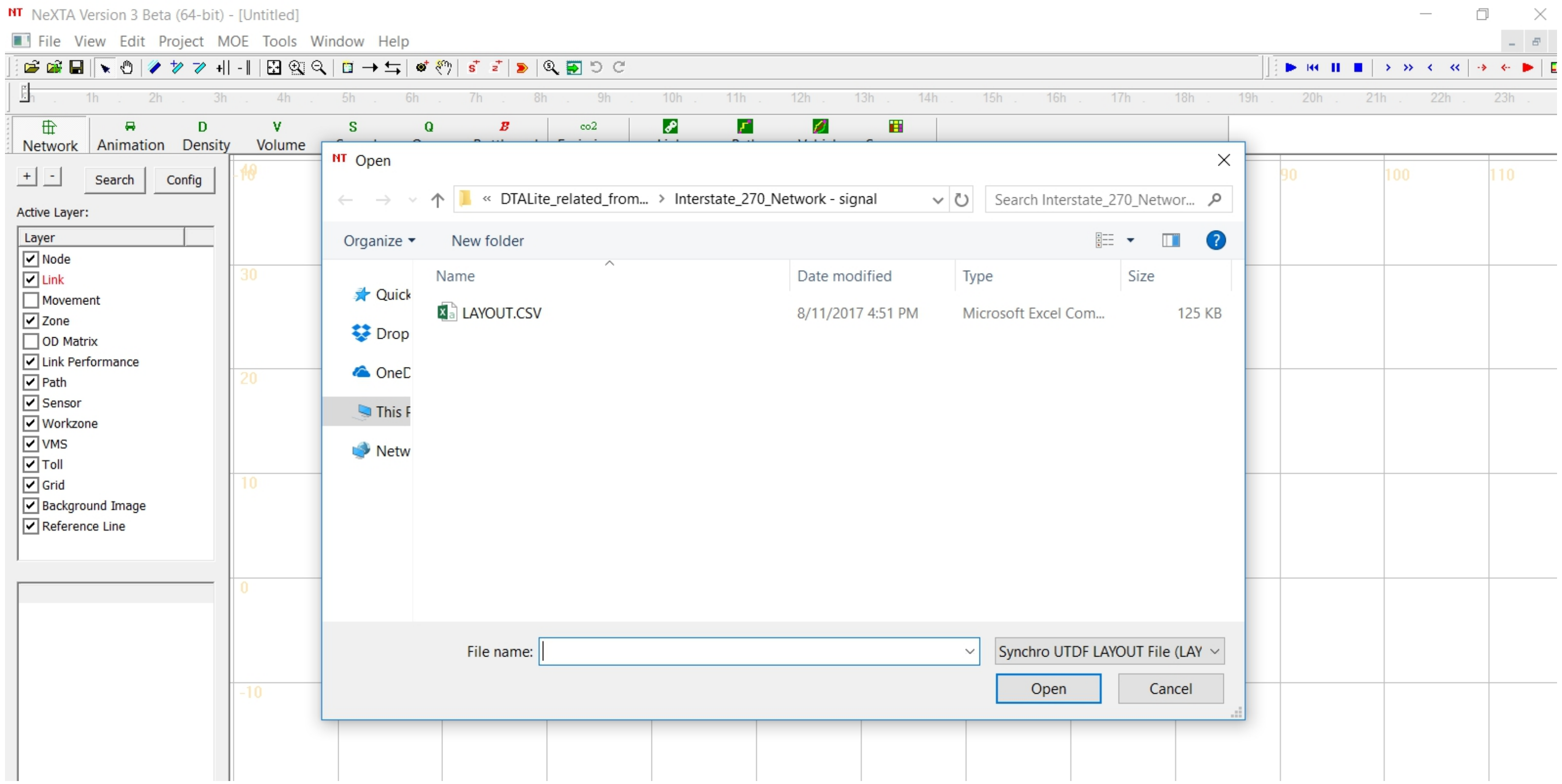
- Open NEXTA, click tools -> Import Synchro UTDF File:



# 7.3 Importing data from Synchro to DTALite

## Step 2: Use NEXTA to import the Synchro UTDF format.

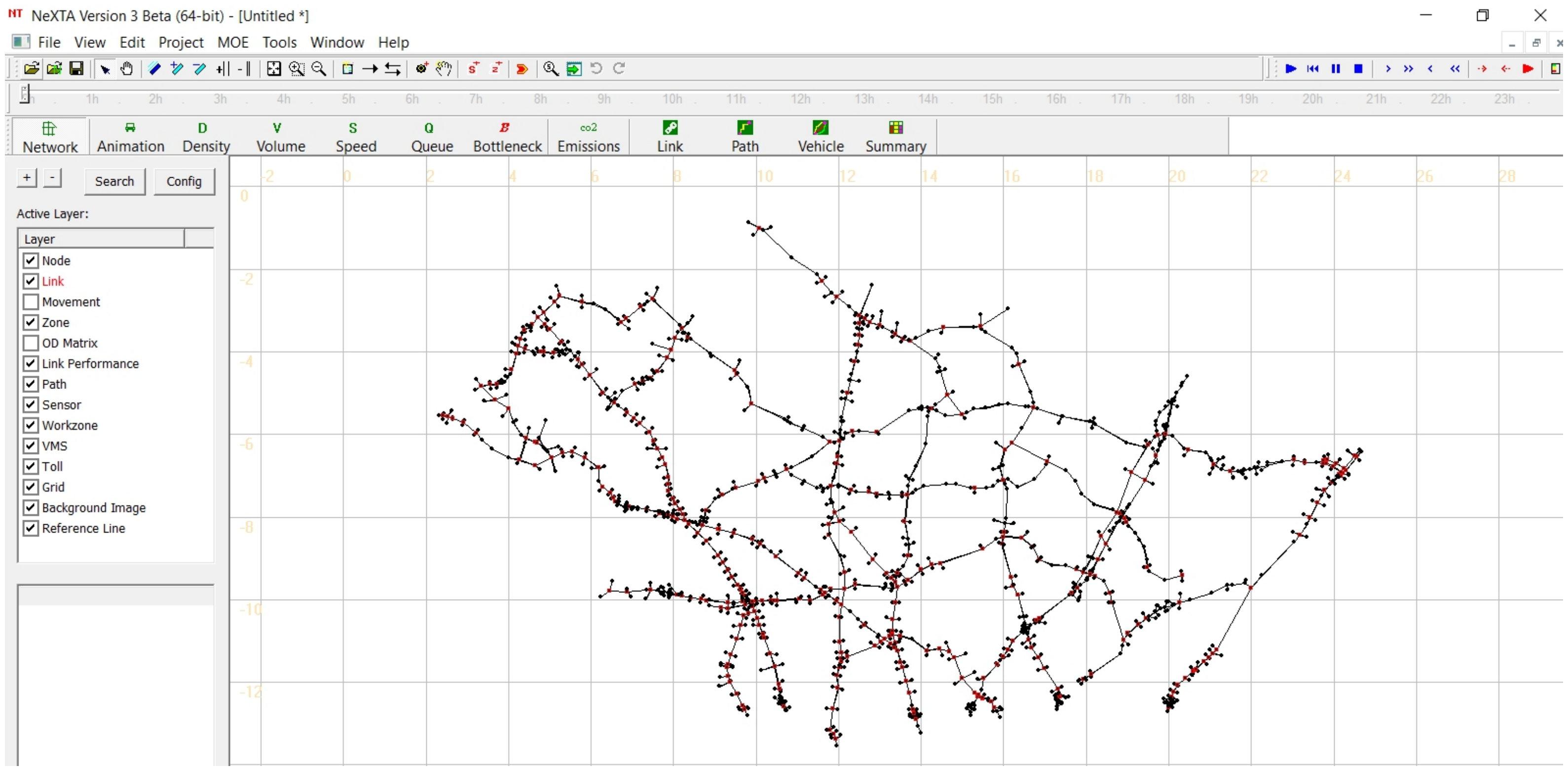
- Open LAYOUT.csv File



# 7.3 Importing data from synchro to DTALite

## Step 2: Use NEXTA to import the Synchro UTDF format.

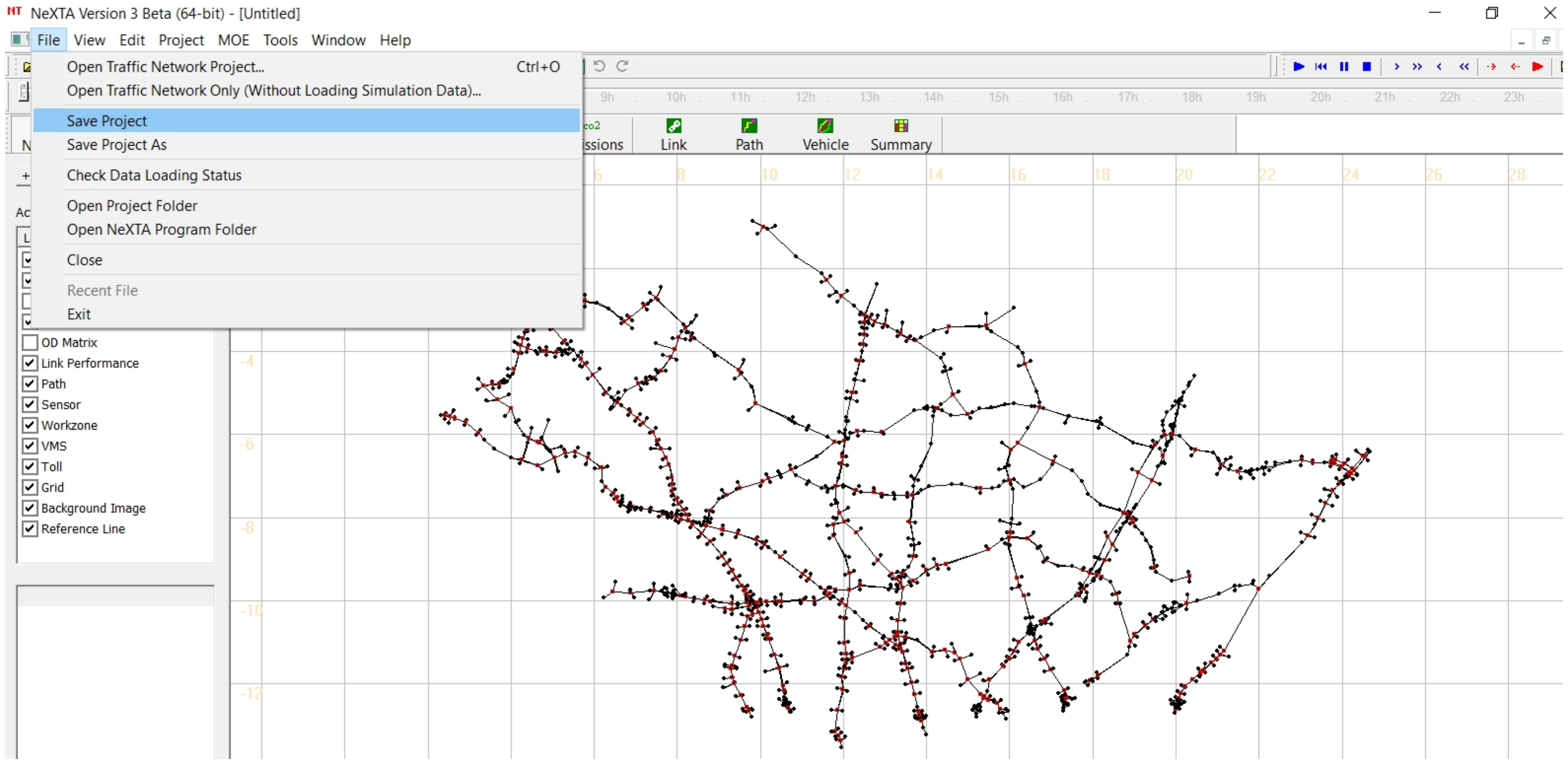
- The synchro data will be imported to NEXTA.



# 7.3 Importing data from synchro to DTALite

## Step 2: Use NEXTA to import the Synchro UTDF format.

- Save the project in the same folder



# 7.3 Importing data from synchro to DTALite

## Step 2: Use NEXTA to import the Synchro UTDF format.

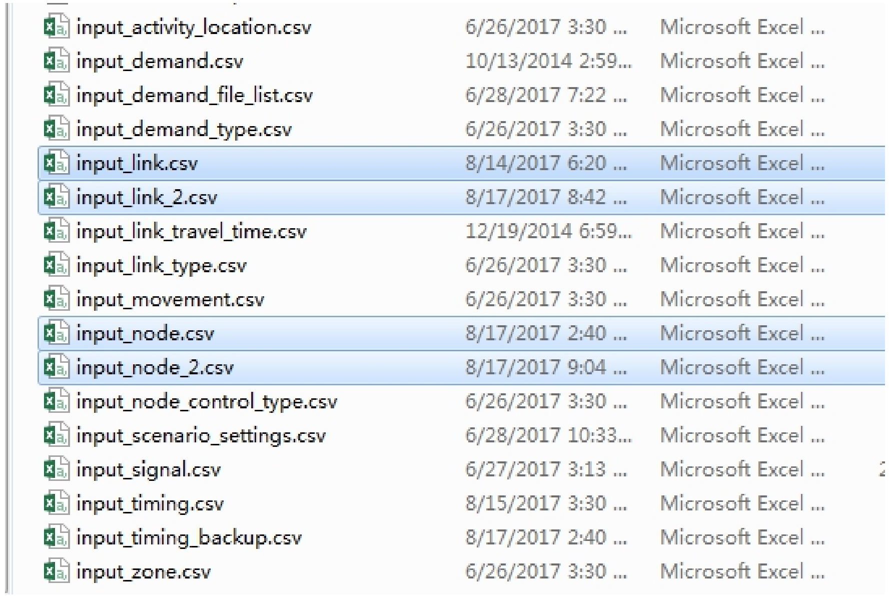
- The network and the signal data will be automatically saved into the folder.

| Name | Date modified | Type | Size |
|---|---|---|---|
| input_activity_location.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| input_demand_type.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| input_link_type.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| input_timing_backup.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 76 KB |
| input_zone.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| output_zone.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| Scenario_Dynamic_Message_Sign.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| Scenario_Link_Based_Toll.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| Scenario_Work_Zone.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| synchro_270.tnp | 8/13/2017 9:30 PM | TNP File | 1 KB |
| input_link.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 827 KB |
| input_movement.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 306 KB |
| input_node.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 167 KB |
| input_node_control_type.csv | 8/13/2017 9:30 PM | Microsoft Excel Com... | 1 KB |
| LANES.CSV | 8/11/2017 4:51 PM | Microsoft Excel Com... | 1,816 KB |
| LAYOUT.CSV | 8/11/2017 4:51 PM | Microsoft Excel Com... | 125 KB |
| PHASING.CSV | 8/11/2017 4:51 PM | Microsoft Excel Com... | 335 KB |
| TIMING.CSV | 8/11/2017 4:51 PM | Microsoft Excel Com... | 17 KB |
| output_LinkMOE.csv | 7/10/2017 2:34 PM | Microsoft Excel Com... | 538 KB |
| output_LinkTDMOE.bin | 7/10/2017 2:34 PM | BIN File | 44,958 KB |
| agent.bin | 7/10/2017 2:34 PM | BIN File | 75,380 KB |
| input_scenario_settings.csv | 6/28/2017 10:33 AM | Microsoft Excel Com... | 1 KB |
| demand.dat | 10/13/2014 2:59 AM | DAT File | 189 KB |
| demand_HOV.dat | 10/13/2014 2:59 AM | DAT File | 189 KB |
| demand_truck.dat | 10/13/2014 2:59 AM | DAT File | 189 KB |

# 7.3 Importing data from synchro to DTALite

## Step 3: Mapping two networks

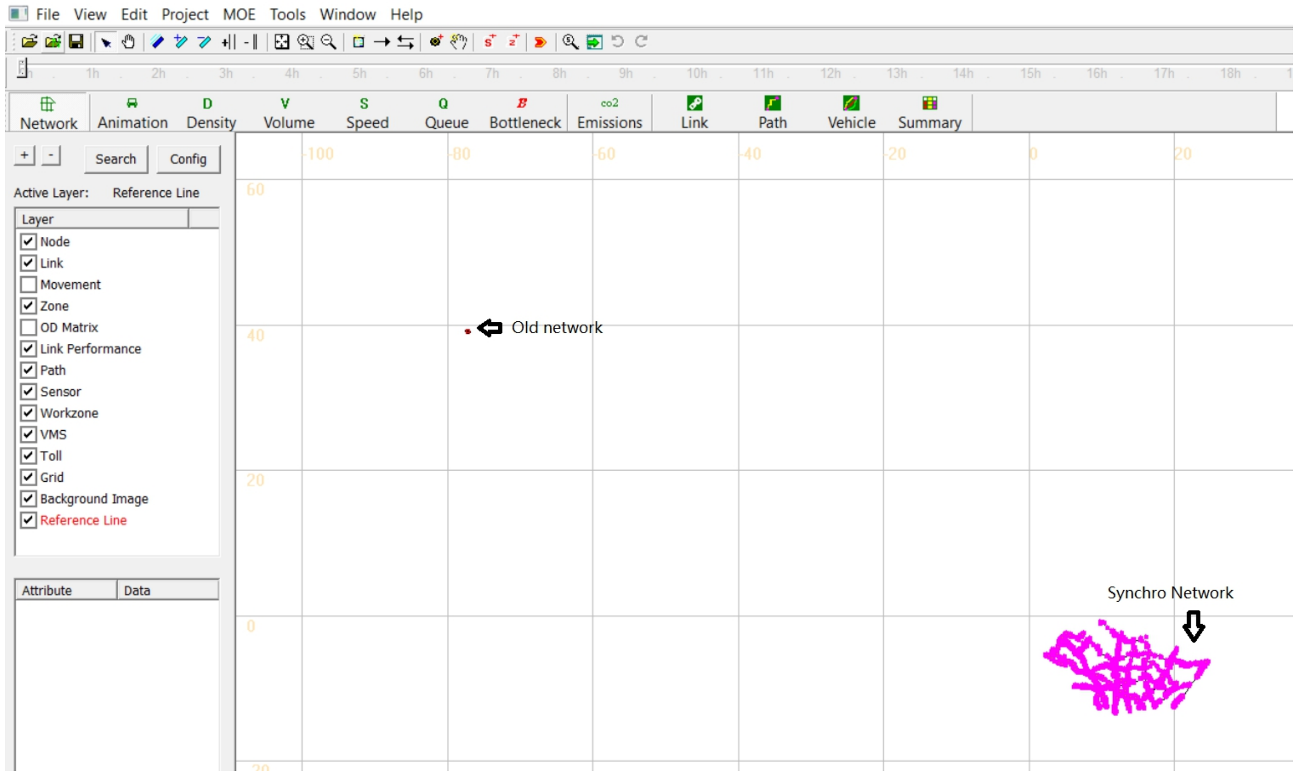
- In this step, we have two networks (a) DTALite main network (b) the imported synchro network in DTALite format.
- Rename the input_node.csv and input_link.csv to input_node_2.csv and input_link_2.csv and copy to our old DTALite network.



# 7.3 Importing data from synchro to DTALite

## Step 3: Mapping two networks

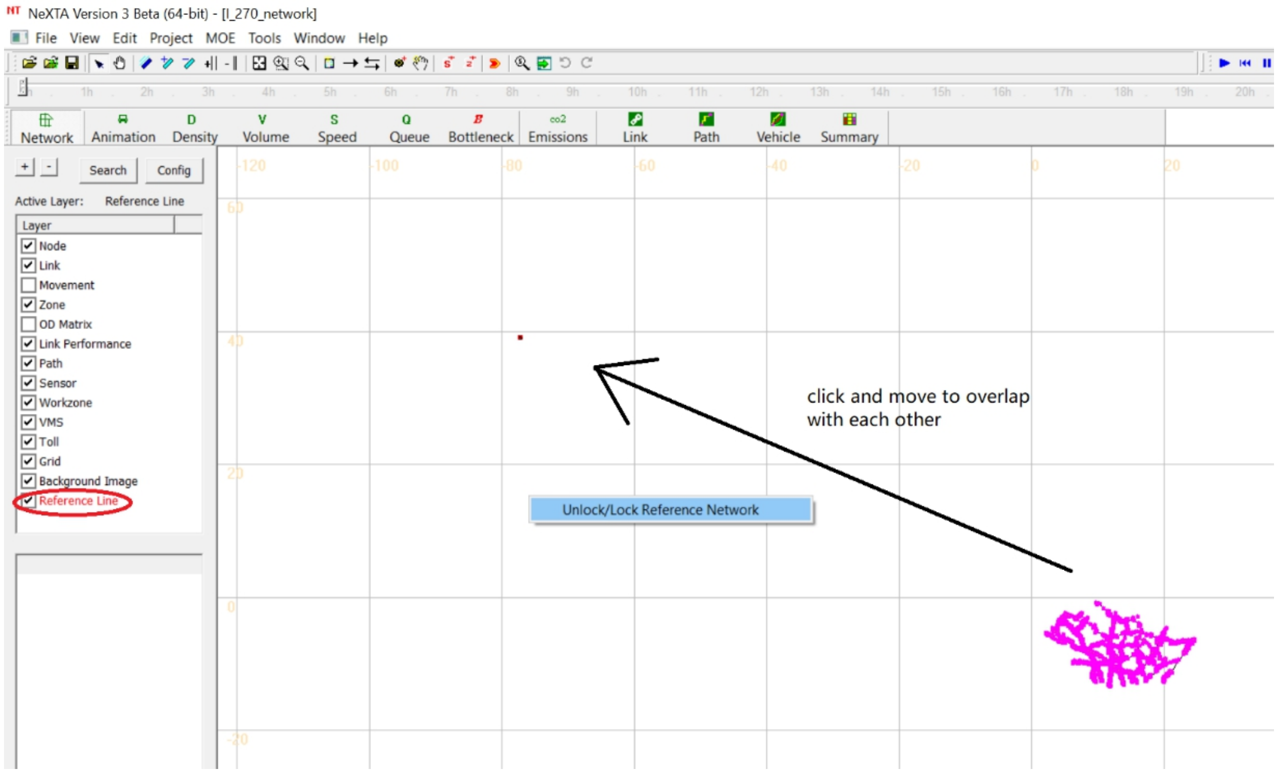
- Then, open the new project in NEXTA.



# 7.3 Importing data from synchro to DTALite

## Step 3: Mapping two networks

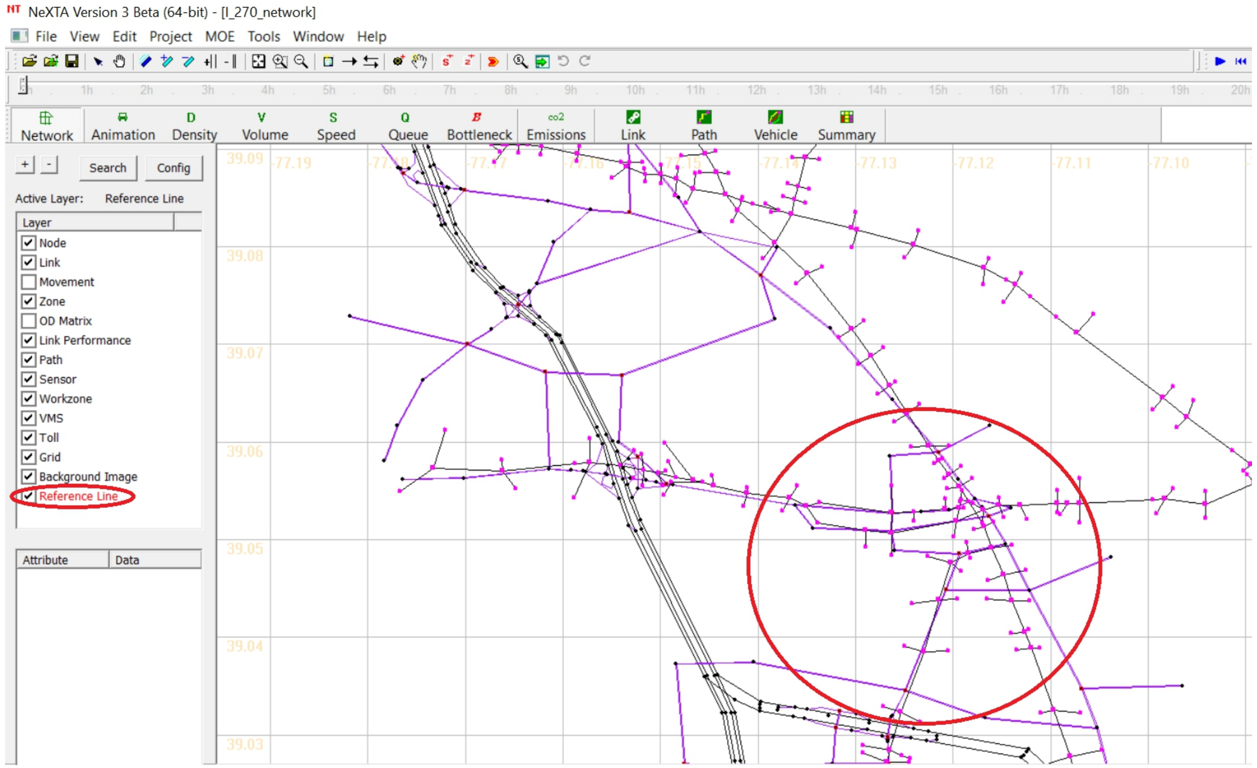
- Use lock/unlock reference network tool in reference line layer



# 7.3 Importing data from synchro to DTALite

## Step 3: Mapping two networks

- Move, zoom in and zoom out imported synchro network



# 7.3 Importing data from synchro to DTALite

## Step 3: Mapping two networks

- But now the node and Link ID of the two networks are different, we need to map them with our old DTALite Network.
- Based on this, we could have the input_node_mapping.csv table generated by ourselves manually.
- (e.g., Node_id_1 is the node id from the old DTALite network, node_id_2 is the mapping node from synchro exported network.)

| node_id_1 | node_id_2 |
|---|---|
| **3346** | 251 |
| **3335** | 1208 |
| **3338** | 1505 |
| **…** | … |

# 7.3 Importing data from synchro to DTALite

## Step 4: Generate the new input_timing_mapping.csv

- Get input_node_mapping.csv file in the last step
- Copy the input_timing_2.csv from the Synchro to the new folder.
- Then we could run the simulation of timing_movement_generater.exe (prepared by Jun Zhao from UMD) and get the result of input_timing_mapping.csv:

| | | | |
|---|---|---|---|
| input_node_mapping.csv | 8/17/2017 6:12 PM | Microsoft Excel Co... | 1 KB |
| input_timing.csv | 8/19/2017 3:55 PM | Microsoft Excel Co... | 1 KB |
| input_timing_2.csv | 8/13/2017 9:30 PM | Microsoft Excel Co... | 76 KB |
| input_timing_mapping.csv | 8/19/2017 4:04 PM | Microsoft Excel Co... | 1 KB |

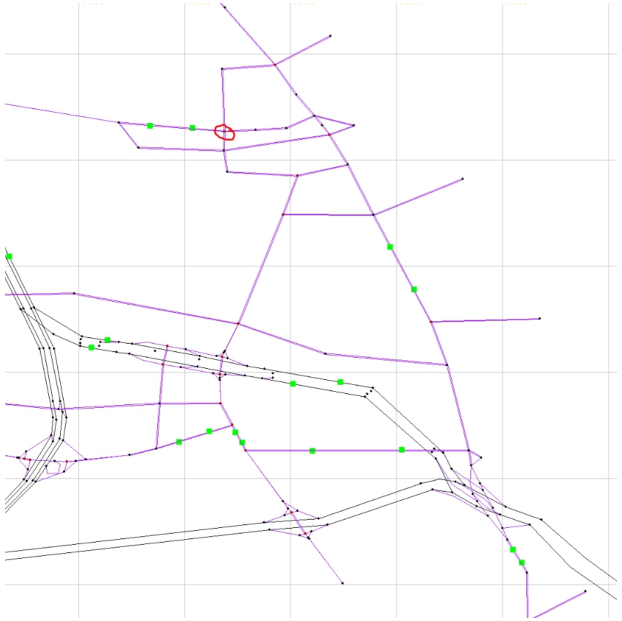
C:\DTALite\DTALite_related_from_DRZhou_0804\Shared signal test data_0820\4-input_timi...

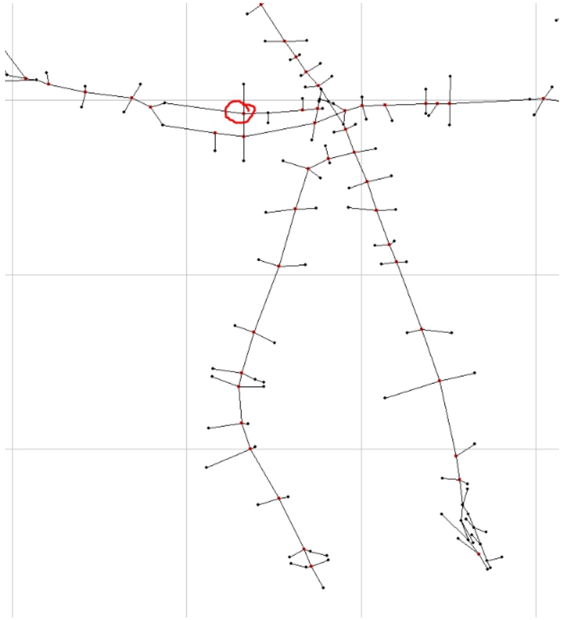
```
reading input_node_mapping.csv ...

the node_id in DTA Network is: 3346
the node_id in Synchro exported Network is: 251
the node_id in DTA Network is: 3335
the node_id in Synchro exported Network is: 1208
reading input_timing.csv ...

reading input_timing2.csv ...

Press any key to continue . . .
```

# 7.3 Importing data from synchro to DTALite

## Step 5: Verify the input phasing results

- Verify the input phasing results and add required next phase number. in main DTA network
- Make sure the imported phasing data are correctly display in NEXTA.

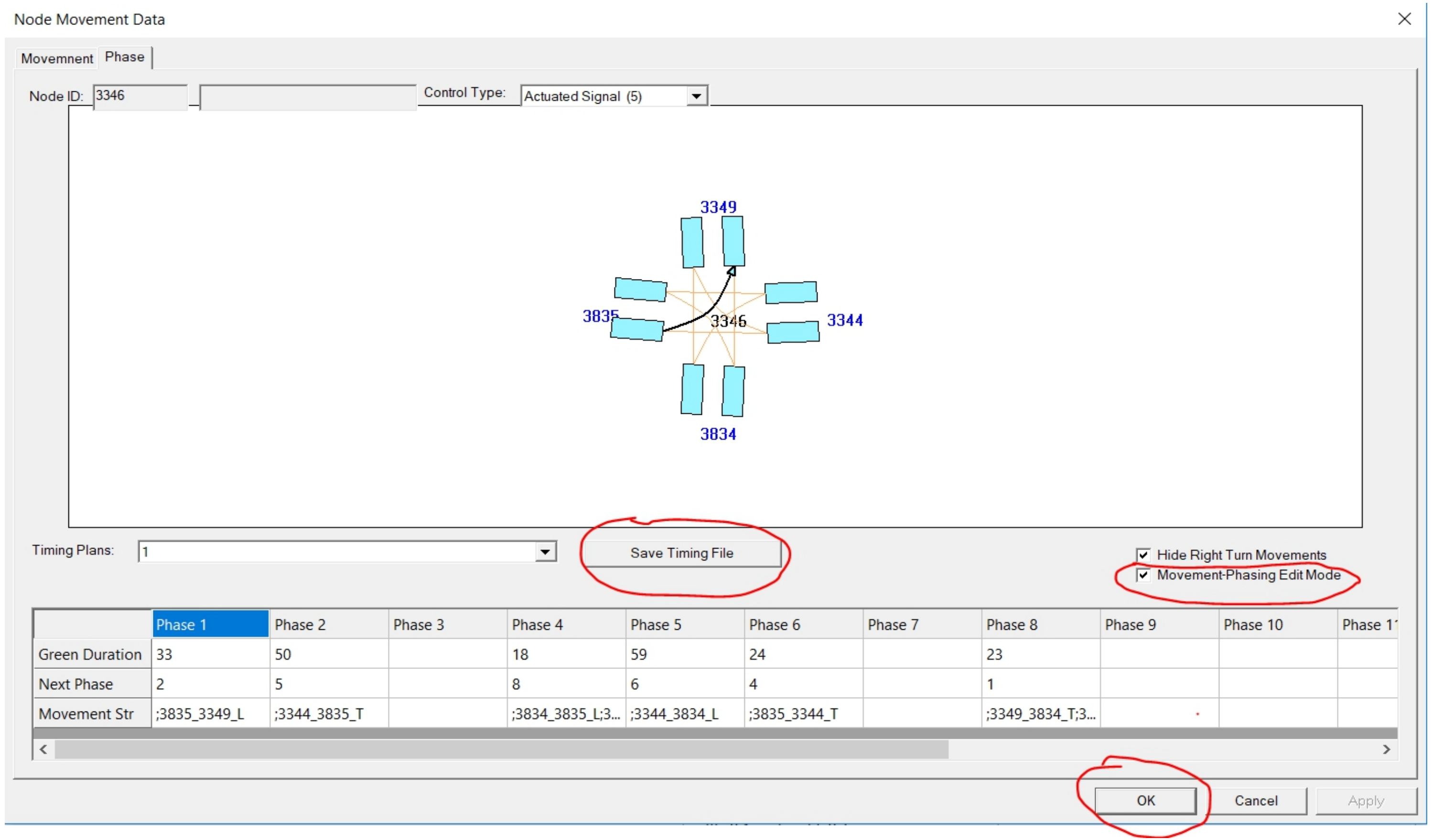




# 7.3 Importing data from synchro to DTALite

## Step 5: Verify the input phasing results

- Manually edit (add, delete or change) the green duration and movement string based on the data exported from Synchro.

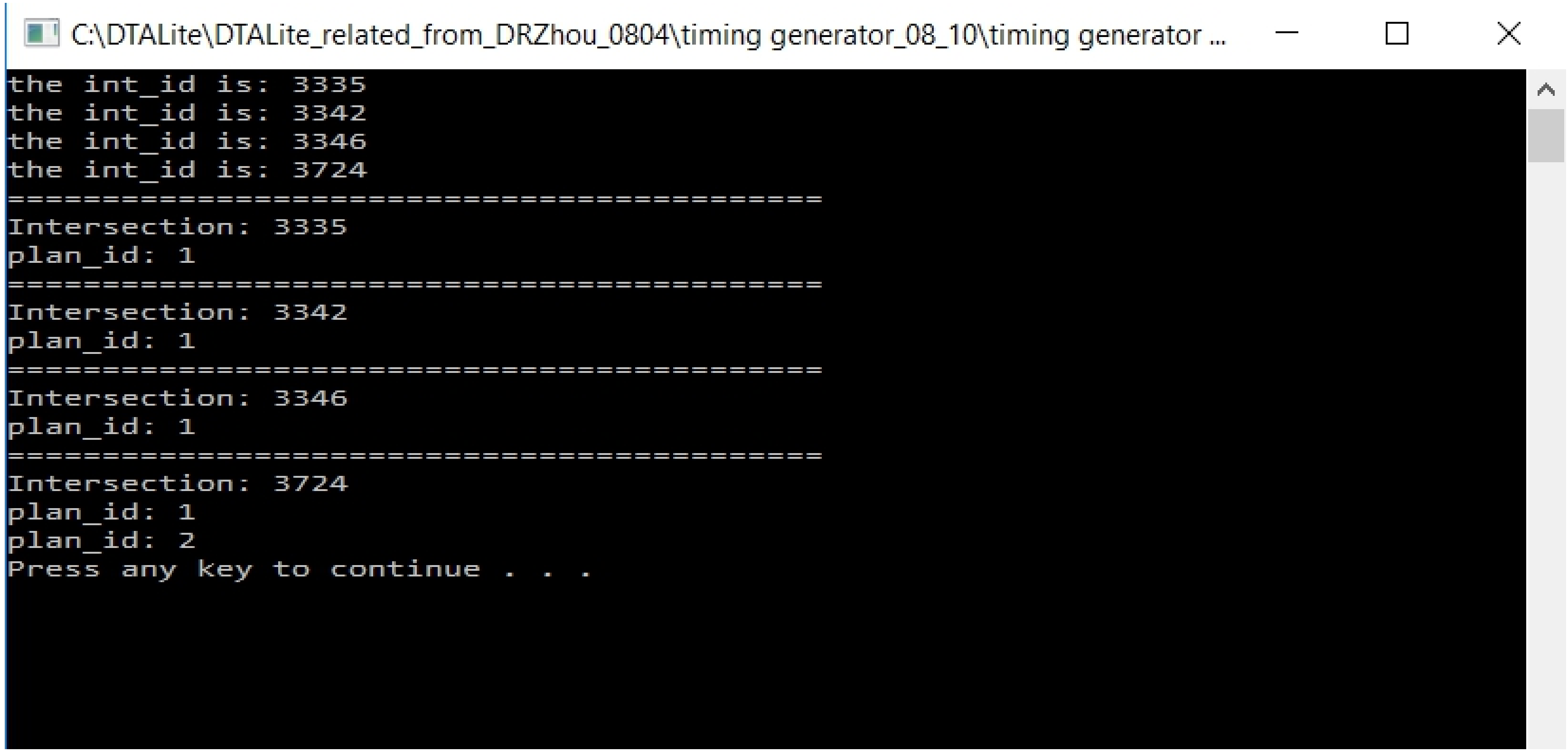



| | Phase 1 | Phase 2 | Phase 3 | Phase 4 | Phase 5 | Phase 6 | Phase 7 | Phase 8 | Phase 9 | Phase 10 | Phase 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Green Duration | 33 | 50 | | 18 | 59 | 24 | | 23 | | | |
| Next Phase | 2 | 5 | | 8 | 6 | 4 | | 1 | | | |
| Movement Str | ;3835_3349_L | ;3344_3835_T | | ;3834_3835_L;3... | ;3344_3834_L | ;3835_3344_T | | ;3349_3834_T;3... | | | |

# 7.3 Importing data from synchro to DTALite

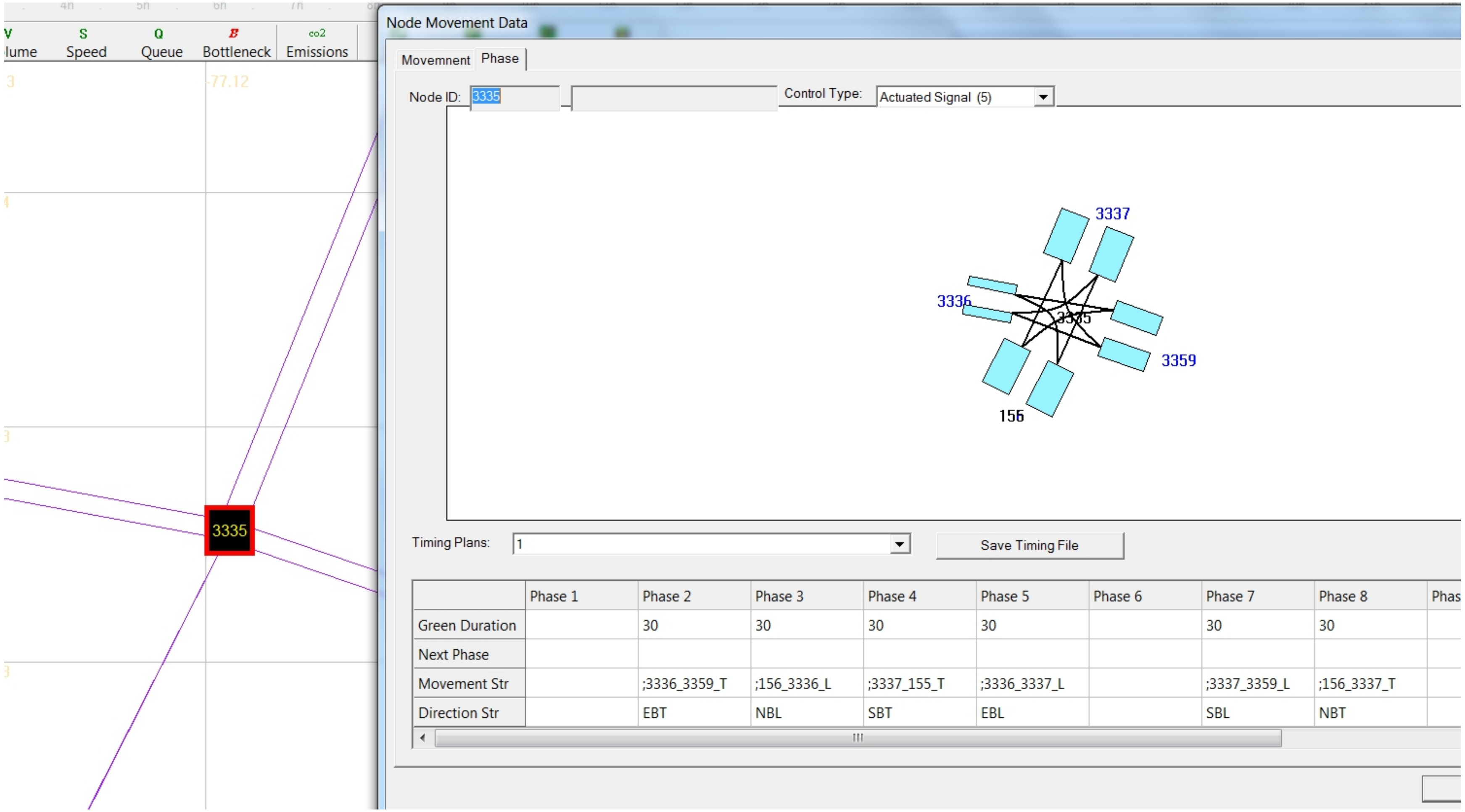
## Step 6: Run timing_generator.exe

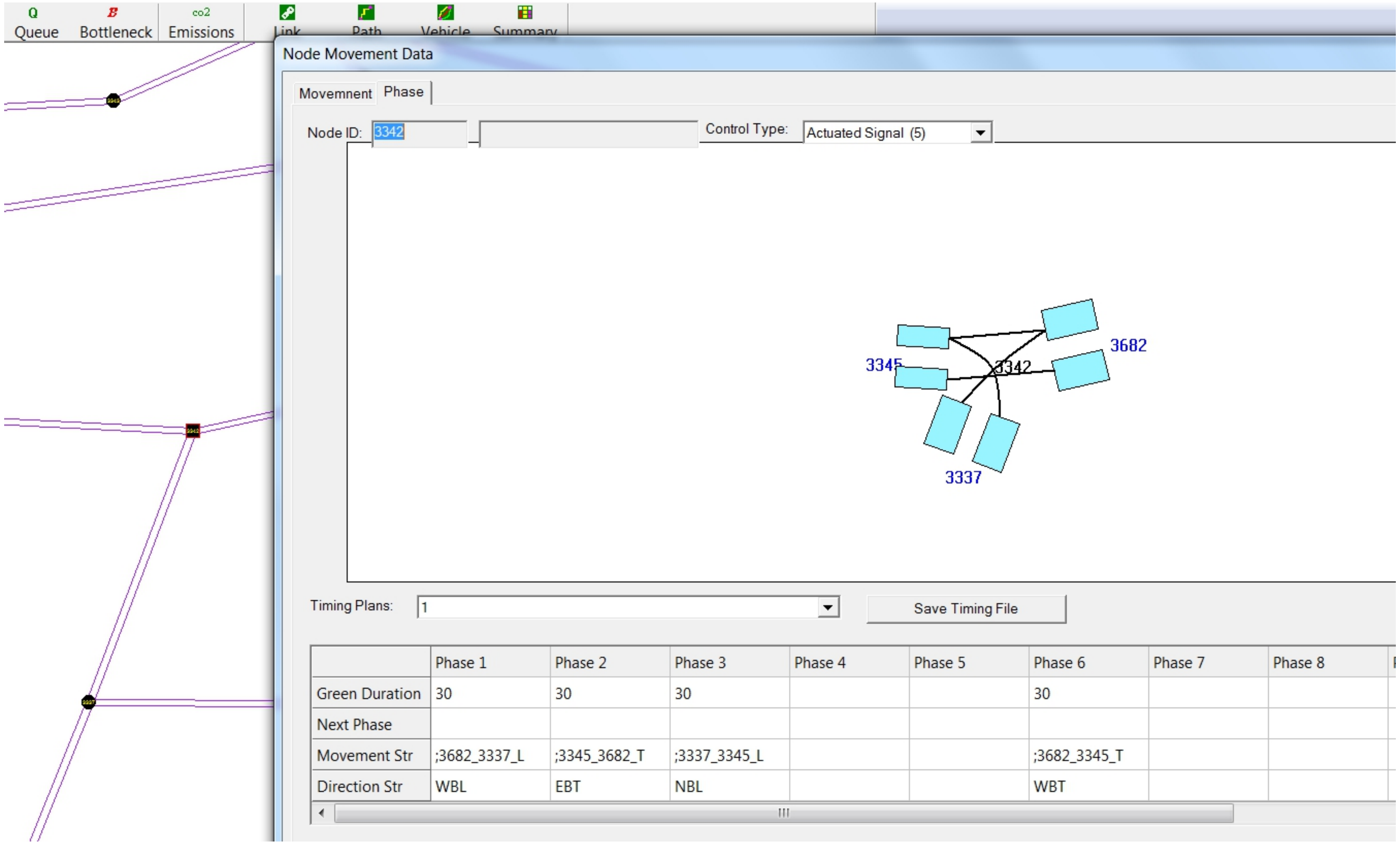
- Change the input_timing_backup.csv file into input_timing.csv and then run timing_generator.exe to get input_timing_status.csv.



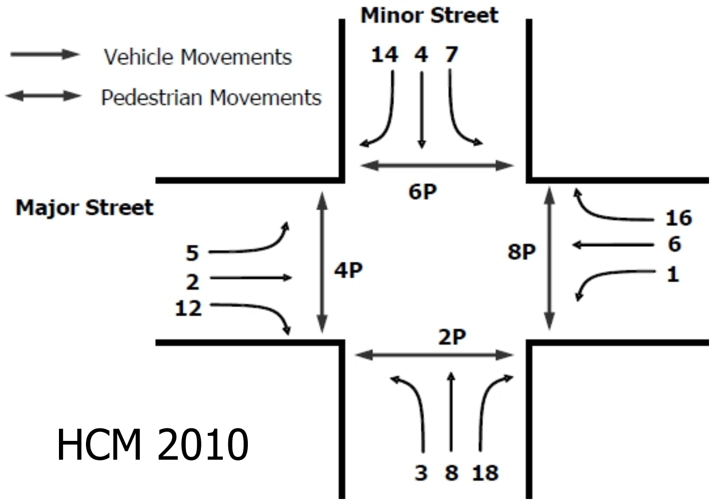
# Generate default NEMA phasing for a signalized intersection

- If a node is designated as a pretimed signal, and there is no input timing information provided, then NeXTA can generate the default NEMA phasing

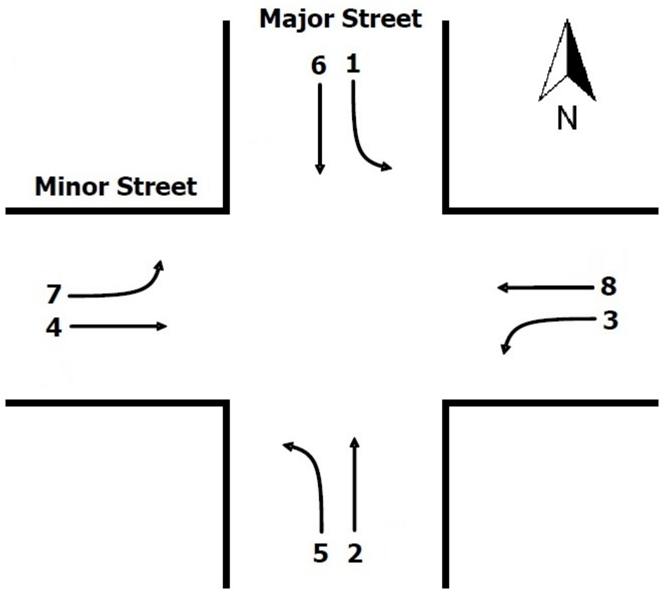

Volume
Speed
Queue
Bottleneck
Emissions
3335
Node Movement Data
Movement
Phase
Node ID: 3335
Control Type: Actuated Signal (5)
3337
3336
3335
3359
156
Timing Plans: 1
Save Timing File
Phase 1 | Phase 2 | Phase 3 | Phase 4 | Phase 5 | Phase 6 | Phase 7 | Phase 8
Green Duration | | 30 | 30 | 30 | 30 | | 30 | 30
Next Phase
Movement Str | | ;3336_3359_T | ;156_3336_L | ;3337_155_T | ;3336_3337_L | | ;3337_3359_L | ;156_3337_T
Direction Str | | EBT | NBL | SBT | EBL | | SBL | NBT


Queue
Bottleneck
Emissions
Link
Path
Vehicle
Summary
Node Movement Data
Movemnent
Phase
Node ID: 3342
Control Type: Actuated Signal (5)
3345
3342
3682
3337
Timing Plans: 1
Save Timing File
Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Phase 6
Phase 7
Phase 8
Green Duration 30 30 30 30
Next Phase
Movement Str ;3682_3337_L ;3345_3682_T ;3337_3345_L ;3682_3345_T
Direction Str WBL EBT NBL WBT

# Movement & Phase Numbering




HCM 2010



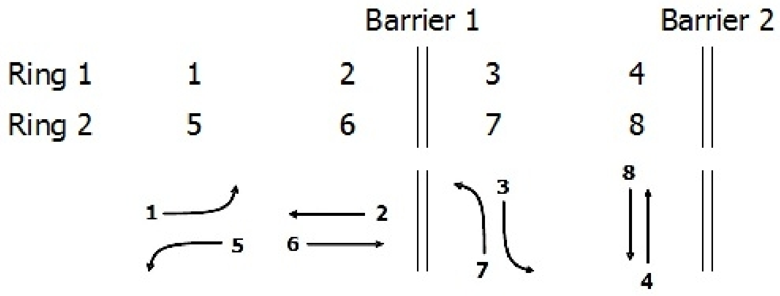

Through Φ: even numbers (2,4,6,8)
Left turn Φ: odd numbers (1,3,5,7)
3-legged intersection LT can be even #

Φ 2 & 6 denote major street
Right turn Φ concurrent w/ through in most cases (no special RT Φ)

Contributing Author:
Dr. Milan Zlatkovic

# Ring-Barrier Phase Structure

- Ring-barrier diagram – graphical representation of phases within sets of rings & barriers (NEMA standard, all traffic controllers)
- Ring – phases that operate in sequence (cannot time simultaneously)
- Barrier - separation of intersecting movements to prevent operating conflicting phases at same time



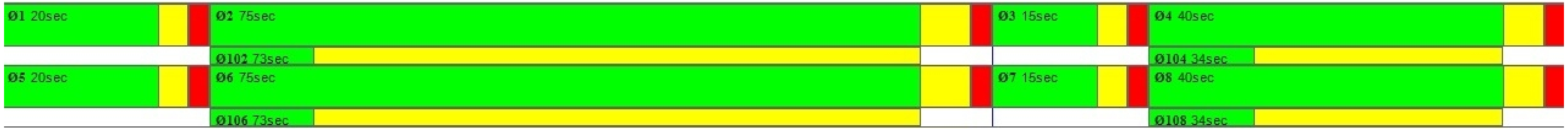

Split summations:

$\Phi\ 1+2+3+4 = C$ $\quad$ $\Phi\ 1+2 = \Phi\ 5+6$

$\Phi\ 5+6+7+8 = C$ $\quad$ $\Phi\ 3+4 = \Phi\ 7+8$

**Exceptions?**

Contributing Author:
Dr. Milan Zlatkovic



# 7.4
# QEM (Signal Quick Estimation Method) Application


Contributing Author:

Dr. Milan Zlatkovic

Assistant Professor, University of Wyoming

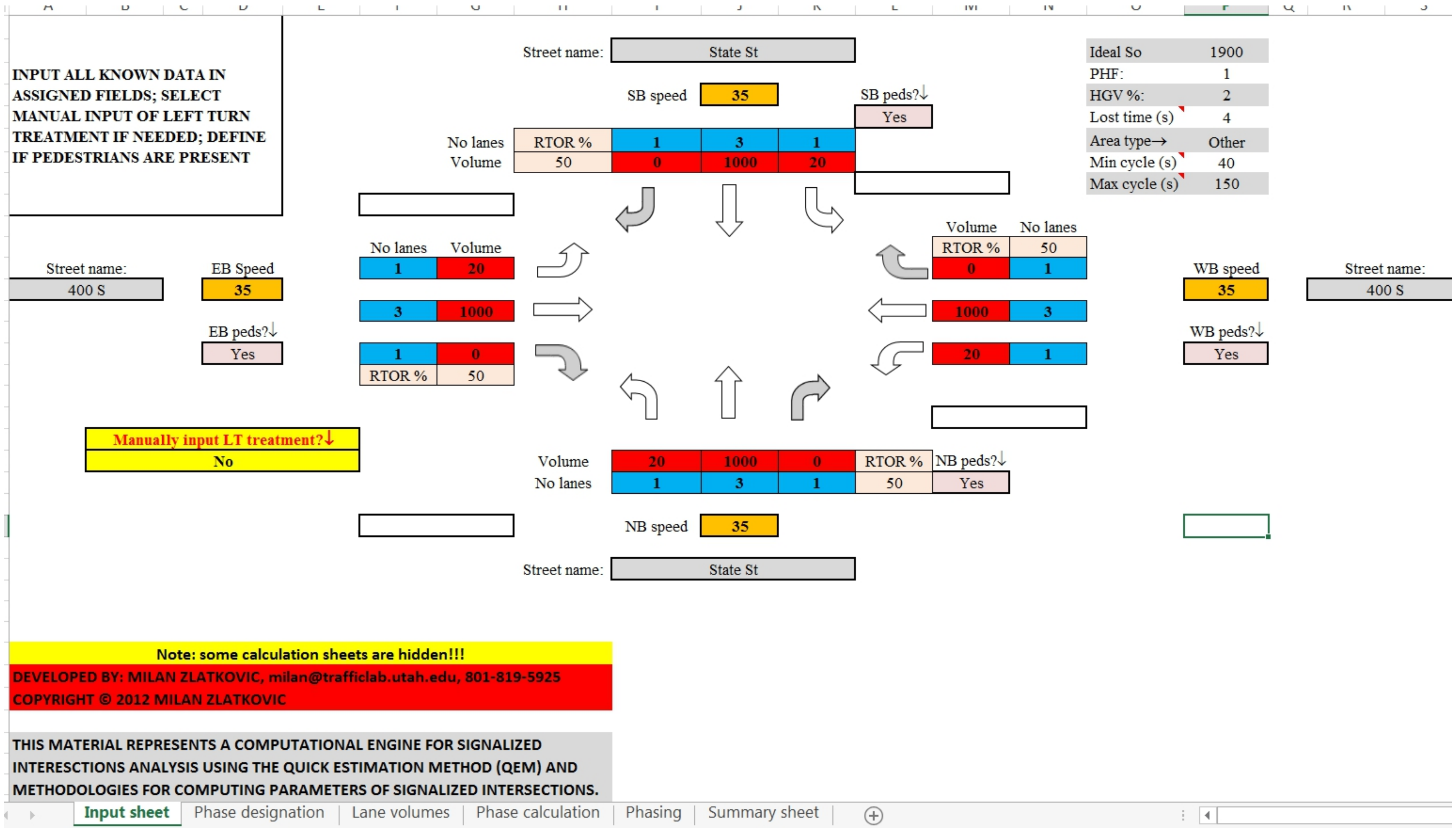
http://www.uwyo.edu/civil/faculty_staff/faculty/milan-zlatkovic/index.html


Reference:


Integration of signal timing estimation model and dynamic traffic assignment in feedback loops: system design and case study

Milan Zlatkovic, Xuesong Zhou

Journal of Advanced Transportation 49 (6), 683-699, 2015


INPUT ALL KNOWN DATA IN ASSIGNED FIELDS; SELECT MANUAL INPUT OF LEFT TURN TREATMENT IF NEEDED; DEFINE IF PEDESTRIANS ARE PRESENT
Street name: State St
SB speed 35
SB peds?↓ Yes
No lanes RTOR % 1 3 1
Volume 50 0 1000 20
Ideal So 1900
PHF: 1
HGV %: 2
Lost time (s) 4
Area type→ Other
Min cycle (s) 40
Max cycle (s) 150
Street name: 400 S
EB Speed 35
EB peds?↓ Yes
No lanes Volume
1 20
3 1000
1 0
RTOR % 50
Volume No lanes
RTOR % 50
0 1
1000 3
20 1
WB speed 35
WB peds?↓ Yes
Street name: 400 S
Manually input LT treatment?↓
No
Volume 20 1000 0 RTOR % NB peds?↓
No lanes 1 3 1 50 Yes
NB speed 35
Street name: State St
Note: some calculation sheets are hidden!!!
DEVELOPED BY: MILAN ZLATKOVIC, milan@trafficlab.utah.edu, 801-819-5925
COPYRIGHT © 2012 MILAN ZLATKOVIC
THIS MATERIAL REPRESENTS A COMPUTATIONAL ENGINE FOR SIGNALIZED INTERESCTIONS ANALYSIS USING THE QUICK ESTIMATION METHOD (QEM) AND METHODOLOGIES FOR COMPUTING PARAMETERS OF SIGNALIZED INTERSECTIONS.
Input sheet
Phase designation
Lane volumes
Phase calculation
Phasing
Summary sheet

# QEM Application: Overview

- Challenges for Planners:
  - How to obtain signal data? Especially for future year with dramatically changed traffic pattern??
  - How do we obtain movement-specific capacity under prevailing traffic condition (e.g. different turning percentage) for a large number (100-500) of intersections?

- The QEM_SIG Excel spreadsheet that accompanies NeXTA represents a computational engine for estimating signalized intersections operation
- It relies on HCM 2010 and additional methodologies for computing parameters of signalized intersections
- Can be used as a stand-alone application, or integrated with NeXTA
- It is an open-source Excel application with friendly GUI for data input and viewing results

# QEM Application: Sections

- Sheet 1 – input/output sheet for communication with NeXTA
- Input sheet – data input section; this is the only section where the user needs to input intersection data
- Phase designation – assigns phases to intersection movements, determines left turn treatment and defines ring-barrier structure
- Lane volumes – calculation of lane volumes (HCM 2010 Chapter 31)
- Phase calculation – calculation of the cycle length, green time (splits) allocations, movement capacities, V/C ratios, and Levels of Service (LOS)
- Phasing – calculates phasing data for correct export to Synchro
- Summary sheet – output sheet with calculation results

# QEM Application: Input Sheet 1/2

- Street names – optional
- Intersection lane configuration – required for stand-alone application; not needed for use with NeXTA
- Turning volumes for each movement – required for stand-alone application; not needed for use with NeXTA
- Approach through speeds – important if the speeds are greater than 45 mph; not needed for use with NeXTA
- Presence of pedestrians – should be enabled for urban intersections
- Manual selection of left turn treatment – optional for stand-alone application; not needed for use with NeXTA
- Right Turn on Red (RTOR %) – 50-75% for exclusive right turn lanes, < 10% for shared

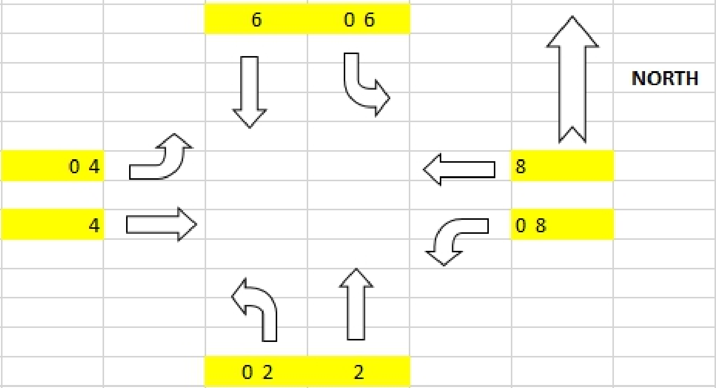
# QEM Application: Input Sheet 2/2

- Ideal saturation flow rate (Ideal So) – default value = 1900 vphpl; can be changed to reflect local conditions
- Peak Hour Factor (PHF) – should be entered if known; otherwise, a default value of 0.9 can be used
- Percentage of heavy vehicles (HGV %) – should be entered for local conditions; otherwise, a default value of 3% can be used
- Lost Time (s) per phase – 4 seconds by default; can be changed to reflect local conditions
- Area Type selection – “CBD” (Central business district) and “Other”; should be selected for the analyzed network
- Minimum and maximum desired cycle lengths – by default, the cycle length is between 40 s (minimum) and 150 s (maximum)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| InitPhase No. | 7 | 4 | 3 | 8 | 5 | 2 | 1 | 6 |
| 3-leg? | 7 | 4 | 3 | 8 | 5 | 2 | 1 | 6 |
| No Lanes | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |
| Shared | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ThruSpeedLimit | | 35 | | 35 | | 35 | | 35 |

| Criteria: | EBL | WBL | NBL | SBL | |
|---|---|---|---|---|---|
| LT lanes | 0 | 0 | 0 | 0 | Prot criteria |
| OppThruLanes | 0 | 0 | 0 | 0 | |
| OppThruSpeed | 0 | 0 | 0 | 0 | |
| LT volume | 0 | 0 | 0 | 0 | |
| Vlt*Vo for prot | 0 | 0 | 0 | 0 | |
| Protected decision | 0 | 0 | 0 | 0 | |
| Vlt*Vo | 2 | 2 | 2 | 2 | perm vs. p |
| ShareLanes | 0 | 0 | 0 | 0 | Split phase |
| 3-leg? | 0 | 0 | 0 | 0 | |
| | | | | | |
| Initial Decision | 2 | 2 | 2 | 2 | |
| Final Decision | 2 | 2 | 2 | 2 | |
| | Permitted | Permitted | Permitted | Permitted | |

| **Movement** | **EBL** | **EBT** | **WBL** | **WBT** | **NBL** | **NBT** | **SBL** | **SBT** |
|---|---|---|---|---|---|---|---|---|
| **Protected Phase** | **0** | **4** | **0** | **8** | **0** | **2** | **0** | **6** |
| **Permitted Phase** | **4** | | **8** | | **2** | | **6** | |




| | | | |
|---|---|---|---|
| Final Ring-Barrier Structure | | | |
| | | b | |
| Ring 1 | | 2 | 4 |
| Ring 2 | | 6 | 8 |

Input sheet | Phase designation | Lane volumes | Phase calculation | Phasing | Summary shee some calculation cells are hidden!

## EB APPROACH

| | | | |
|---|---|---|---|
| **RIGHT TURNS** | Exclusive RT lar | Shared RT lane | |
| RT volume, Vr (vph) | 0 | 0 | |
| No of excl. RT lanes, Nrt | 1 | 1 | |
| RT adj factor, frt | 0.85 | 0.85 | |
| RT vol per lane, Vrt (vphpl) | 0 | 0 | |
| | | | |
| **LEFT TURNS** | | | |
| LT volume, Vl (vph) | 20 | | |
| Opposing mainline vol, Vo (vph) | 1000 | | |
| No of excl. LT lanes, Nlt | 1 | | |
| LT adj factor, flt | 0.95 | 0 | |
| LT vol per lane Vlt (vphpl) | Permitted LT | Protected LT | Not Opposed LT |
| | 0 | 0 | 0 |
| | | 21 | |
| single lane, regular intersection | 0.95 | | |
| >=2 lanes, regular int. | 0 | | |
| single lane, 1-way or 3-leg int. | 0 | | |
| >=2 lanes, 1-way or 3-leg int. | 0 | | |
| | | | |
| **THROUGH MOVEMENT** | Permitted LT | Protected LT | Not Opposed LT |
| Through volume, Vt (vph) | 1000 | 0 | 0 |
| Parking adjustment factor, fp | 1 | 1 | 1 |
| Number of through lanes, Nth | 3 | 3 | 3 |
| Total approach volume, Vtot (vph) | 1000 | 0 | 0 |
| | | | |
| **THROUGH MOVEMENT WITH EXCLUSIVE LT LANE** | Permitted LT | Protected LT | Not Opposed LT |
| Through volume per lane, Vth (vph | 333 | 0 | 0 |
| Critical lane volume,Vcl (vph) Max[Vlt, Vrt (exclusive), Vth] | 333 | 0 | 0 |
| | | | |
| **THROUGH MOVEMENT WITH SHARED LT LANE** | Permitted LT | Protected LT | Not Opposed LT |
| Proportion of left turns, Plt | 1 | 0 | 0 |
| Equivalence factor, El1 | 4 | 0 | 0 |
| Shared lane LT adjustment factor, f | 1 | 1 | 1 |
| Through volume per lane, Vth (vph | 0 | 0 | 0 |
| Critical lane volume,Vcl (vph) Max[Vrt (exclusive), Vth] | 0 | 0 | 0 |

## WB APPROACH

| | | | |
|---|---|---|---|
| **RIGHT TURNS** | Exclusive RT lar | Shared RT lane | |
| RT volume, Vr (vph) | 0 | 0 | |
| No of excl. RT lanes, Nrt | 1 | 1 | |
| RT adj factor, frt | 0.85 | 0.85 | |
| RT vol per lane, Vrt (vphpl) | 0 | 0 | |
| | | | |
| **LEFT TURNS** | | | |
| LT volume, Vl (vph) | 20 | | |
| Opposing mainline vol, Vo (vph) | 1000 | | |
| No of excl. LT lanes, Nlt | 1 | | |
| LT adj factor, flt | 0.95 | 0 | |
| LT vol per lane Vlt (vphpl) | Permitted LT | Protected LT | Not Opposed LT |
| | 0 | 0 | 0 |
| | | 21 | |
| single lane, regular intersection | 0.95 | | |
| >=2 lanes, regular int. | 0 | | |
| single lane, 1-way or 3-leg int. | 0 | | |
| >=2 lanes, 1-way or 3-leg int. | 0 | | |
| | | | |
| **THROUGH MOVEMENT** | Permitted LT | Protected LT | Not Opposed LT |
| Through volume, Vt (vph) | 1000 | 0 | 0 |
| Parking adjustment factor, fp | 1 | 1 | 1 |
| Number of through lanes, Nth | 3 | 3 | 3 |
| Total approach volume, Vtot (vph) | 1000 | 0 | 0 |
| | | | |
| **THROUGH MOVEMENT WITH EXCLUSIVE LT LANE** | Permitted LT | Protected LT | Not Opposed LT |
| Through volume per lane, Vth (vphp | 333 | 0 | 0 |
| Critical lane volume,Vcl (vph) Max[Vlt, Vrt (exclusive), Vth] | 333 | 0 | 0 |
| | | | |
| **THROUGH MOVEMENT WITH SHARED LT LANE** | Permitted LT | Protected LT | Not Opposed LT |
| Proportion of left turns, Plt | 1 | 0 | 0 |
| Equivalence factor, El1 | 4 | 0 | 0 |
| Shared lane LT adjustment factor, fi | 1 | 1 | 1 |
| Through volume per lane, Vth (vphp | 0 | 0 | 0 |
| Critical lane volume,Vcl (vph) Max[Vrt (exclusive), Vth] | 0 | 0 | 0 |

## NB APPROACH

| | | |
|---|---|---|
| **RIGHT TURNS** | Exclusive RT lar | Sharec |
| RT volume, Vr (vph) | 0 | |
| No of excl. RT lanes, Nrt | 1 | |
| RT adj factor, frt | 0.85 | ( |
| RT vol per lane, Vrt (vphpl) | 0 | |
| | | |
| **LEFT TURNS** | | |
| LT volume, Vl (vph) | 20 | |
| Opposing mainline vol, Vo (vph) | 1000 | |
| No of excl. LT lanes, Nlt | 1 | |
| LT adj factor, flt | 0.95 | |
| LT vol per lane Vlt (vphpl) | Permitted LT | Prote |
| | 0 | |
| single lane, regular intersection | 0.95 | |
| >=2 lanes, regular int. | 0 | |
| single lane, 1-way or 3-leg int. | 0 | |
| >=2 lanes, 1-way or 3-leg int. | 0 | |
| | | |
| **THROUGH MOVEMENT** | Permitted LT | Prote |
| Through volume, Vt (vph) | 1000 | |
| Parking adjustment factor, fp | 1 | |
| Number of through lanes, Nth | 3 | |
| Total approach volume, Vtot (vph) | 1000 | |
| | | |
| **THROUGH MOVEMENT WITH EXCLUSIVE LT LANE** | Permitted LT | Prote |
| Through volume per lane, Vth (vphp | 333 | |
| Critical lane volume,Vcl (vph) Max[Vlt, Vrt (exclusive), Vth] | 333 | |
| | | |
| **THROUGH MOVEMENT WITH SHARED LT LANE** | Permitted LT | Prote |
| Proportion of left turns, Plt | 1 | |
| Equivalence factor, El1 | 4 | |
| Shared lane LT adjustment factor, fi | 1 | |
| Through volume per lane, Vth (vphp | 0 | |
| Critical lane volume,Vcl (vph) Max[Vrt (exclusive), Vth] | 0 | |

some calculation cells are hidden!

| Movement | EBL | EBT | EBR | WBL | WBT | WBR | NBL | NBT | NBR | SBL | SBT | SBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TurnVol | 20 | 1000 | 0 | 21 | 1000 | 0 | 20 | 1000 | 0 | 21 | 1000 | 0 |
| PhF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CompVol | 20 | 1000 | 0 | 21 | 1000 | 0 | 20 | 1000 | 0 | 21 | 1000 | 0 |
| NoLanes | 1 | 3 | 1 | 1 | 3 | 1 | 1 | 3 | 1 | 1 | 3 | 1 |
| RTORreduction | | | 0 | | | 0 | | | 0 | | | 0 |
| LaneGroupFlow | 20 | 1000 | 0 | 21 | 1000 | 0 | 20 | 1000 | 0 | 21 | 1000 | 0 |
| Flu | 1 | 0.908 | | 1 | 0.908 | | 1 | 0.908 | | 1 | 0.908 | |
| SatFlowRateProt | 0 | 4590 | 1530 | 0 | 4590 | 1530 | 0 | 4590 | 1530 | 0 | 4590 | 1530 |
| SatFlowRatePerm | 561 | 4590 | 1530 | 561 | 4590 | 1530 | 561 | 4590 | 1530 | 561 | 4590 | 1530 |
| LaneVolume | 0 | 333 | 0 | 0 | 333 | 0 | 0 | 333 | 0 | 0 | 333 | 0 |
| Prot Phase | 0 | 4 | | 0 | 8 | | 0 | 2 | | 0 | 6 | |
| Perm phase | 4 | 0 | 4 | 8 | 0 | 4 | 2 | 0 | 2 | 6 | 0 | 6 |
| Lost time | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

| | | | |
|---|---|---|---|
| Sum of Lost Time | 8 | | |
| Cycle Calculated | 14.2 | Min Cycle | 70 |
| Cycle Adopted | 70 | | |

| Movement | EBL | EBT | EBR | WBL | WBT | WBR | NBL | NBT | NBR | SBL | SBT | SBR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Phase | 0 | 4 | 4 | 0 | 8 | 8 | 0 | 2 | 2 | 0 | 6 | 6 |
| CalcSplit | 0 | 35 | 35 | 0 | 35 | 35 | 0 | 35 | 35 | 0 | 35 | 35 |
| Yellow | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 |
| AllRed | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 | 2.1 |
| DispGreen | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 | 29.3 |
| EffectiveGren | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 | 31 |
| Green Ratio (g/c) | 0.00 | 0.50 | 0.50 | 0.00 | 0.50 | 0.50 | 0.00 | 0.50 | 0.50 | 0.00 | 0.50 | 0.50 |
| V/C | 0.17 | 0.49 | 0.00 | 0.17 | 0.49 | 0.00 | 0.17 | 0.49 | 0.00 | 0.17 | 0.49 | 0.00 |
| Cap (vph) | 121 | 2033 | 678 | 121 | 2033 | 678 | 121 | 2033 | 678 | 121 | 2033 | 678 |
| ControlDelay (s) | 14.8 | 14.7 | 0.0 | 14.8 | 14.7 | 0.0 | 14.8 | 14.7 | 0.0 | 14.8 | 14.7 | 0.0 |
| LOS | B | B | N/A | B | B | N/A | B | B | N/A | B | B | N/A |
| Approach Delay (s) | | 14.7 | | | 14.7 | | | 14.7 | | | 14.7 | |
| Approach LOS | | B | | | B | | | B | | | B | |

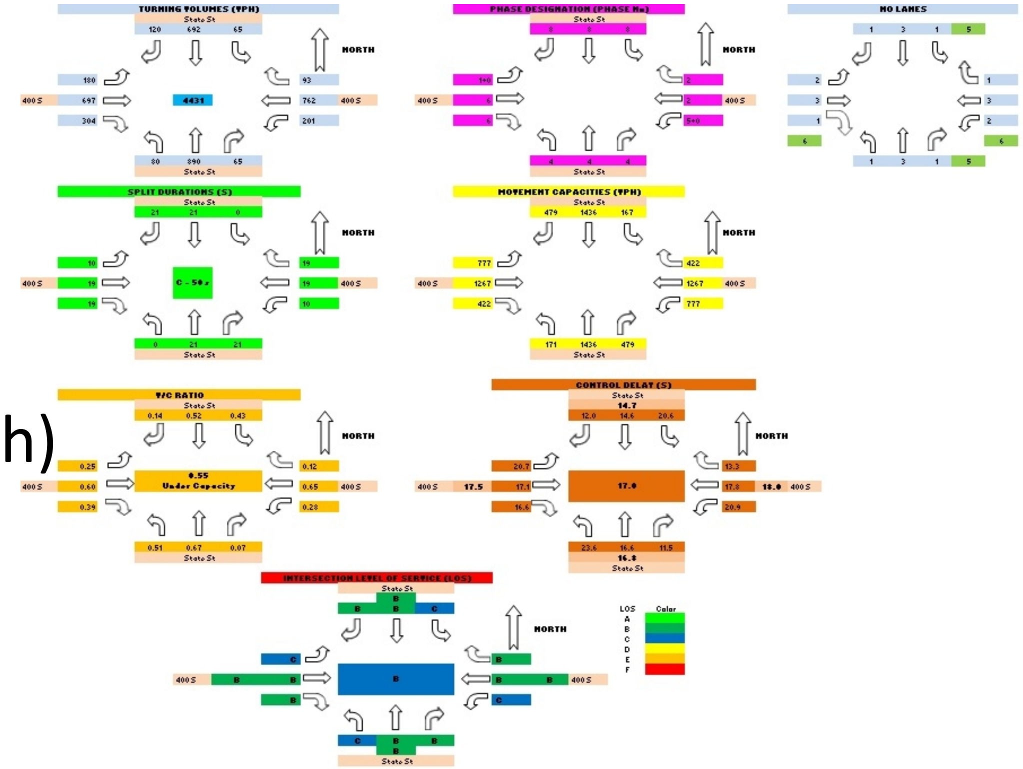
| LOS | Color |
|---|---|
| A | |
| B | |
| C | |
| D | |
| E | |
| F | |

| | |
|---|---|
| **Intersection Delay (s)** | 14.7 |
| **Intersection LOS** | B |
| **Intersection V/C** | 0.44 |
| **Intersection Status** | Under Capacity |

| | | b | | | C check | B1 check | B2 check |
|---|---|---|---|---|---|---|---|
| Ring 1 | 0 | 35 | 0 | 35 | 70 | 35 | 35 |
| Ring 2 | 0 | 35 | 0 | 35 | 70 | 35 | 35 |

# QEM Application: Summary Sheet

- Turning volumes (vph)
- Phase designations
- Number of lanes
- Split durations (s)
- Movement capacities (vph)
- V/C ratios
- Control delays (s)
- Intersection LOS
- Phase data table




| Phase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Movement | EBL | WBT | Not used | NBT | WBL | EBT | Not used | SBT |
| Split (s) | 10 | 19 | Not used | 21 | 10 | 19 | Not used | 21 |
| Capacity (v | 777 | 1267 | Not used | 1436 | 777 | 1267 | Not used | 1436 |
| V/C | 0.25 | 0.65 | Not used | 0.67 | 0.28 | 0.6 | Not used | 0.52 |
| Delay (s) | 20.73 | 17.82 | Not used | 16.6 | 20.93 | 17.05 | Not used | 14.58 |
| LOS | C | B | Not used | B | C | B | Not used | B |

Fig. 20 QEM Summary Sheet

# Module 8: Vehicle Routing Program Source Code

Source code link:

https://github.com/xzhou99/Agent-Plus/blob/master/cpp_source_code_v2_VRP/AgentPlus_VRP/AgentPlus/AgentPlus.cpp

Transportation Research Part B: Methodological

Volume 89, July 2016, Pages 19–42

# Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations

Monirehalsadat Mahmoudi, Xuesong Zhou

Show more

https://doi.org/10.1016/j.trb.2016.03.009



## Highlights

- Model vehicle routing with real-world transportation networks and time-dependent link travel time.
- Develop an efficient dynamic programming algorithm based on state–space–time networks.
- Model pickup and delivery time window constraints using multi-dimensional network flow program.
- Synchronize vehicle routing and determine request pricing within Lagrangian relaxation framework.

## Introduction

### VRP

The vehicle routing problem (VRP) involves finding a set of routes, starting and ending at a depot, that together cover a set of customers. Each customer has a given demand, and no vehicle can service more customers than its capacity permits. The objective is to minimize the total distance traveled or the number of vehicles used, or a combination of these.

### VRPTW

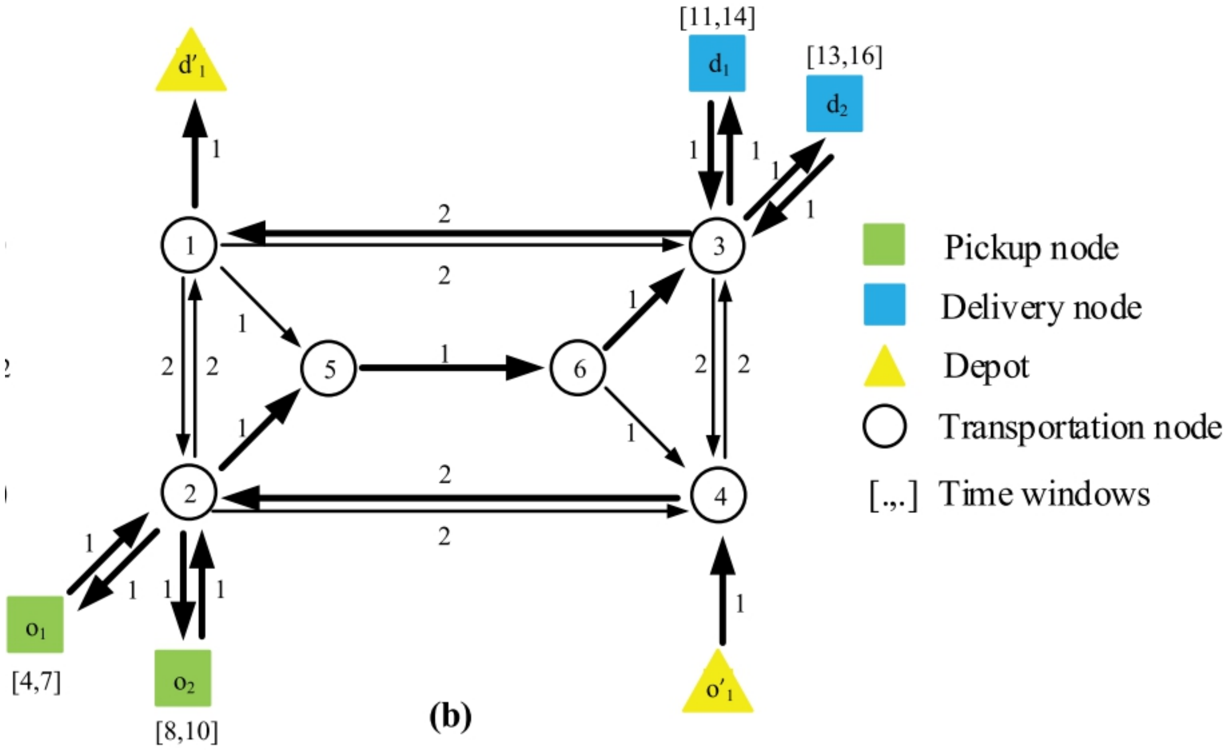
It is a generalization of the VRP where the service at any customer starts within a given time interval, called a time window. Time windows are called soft when they can be considered non-biding for a penalty cost. They are hard when they cannot be violated, i.e., if a vehicle arrives too early at a customer, it must wait until the time window opens; and it is not allowed to arrive late.

**Given**: A set of transportation *requests* and a fleet of *vehicles*

**Task:** Determine a set of vehicle routes to perform all (or some) transportation requests with the given vehicle fleet at minimum cost; in particular, decide which vehicle handles which requests in which sequence so that all vehicle routes can be feasibly executed

## Introduction

### Nodes in VRP




(b)

# Introduction

## Arcs in VRP

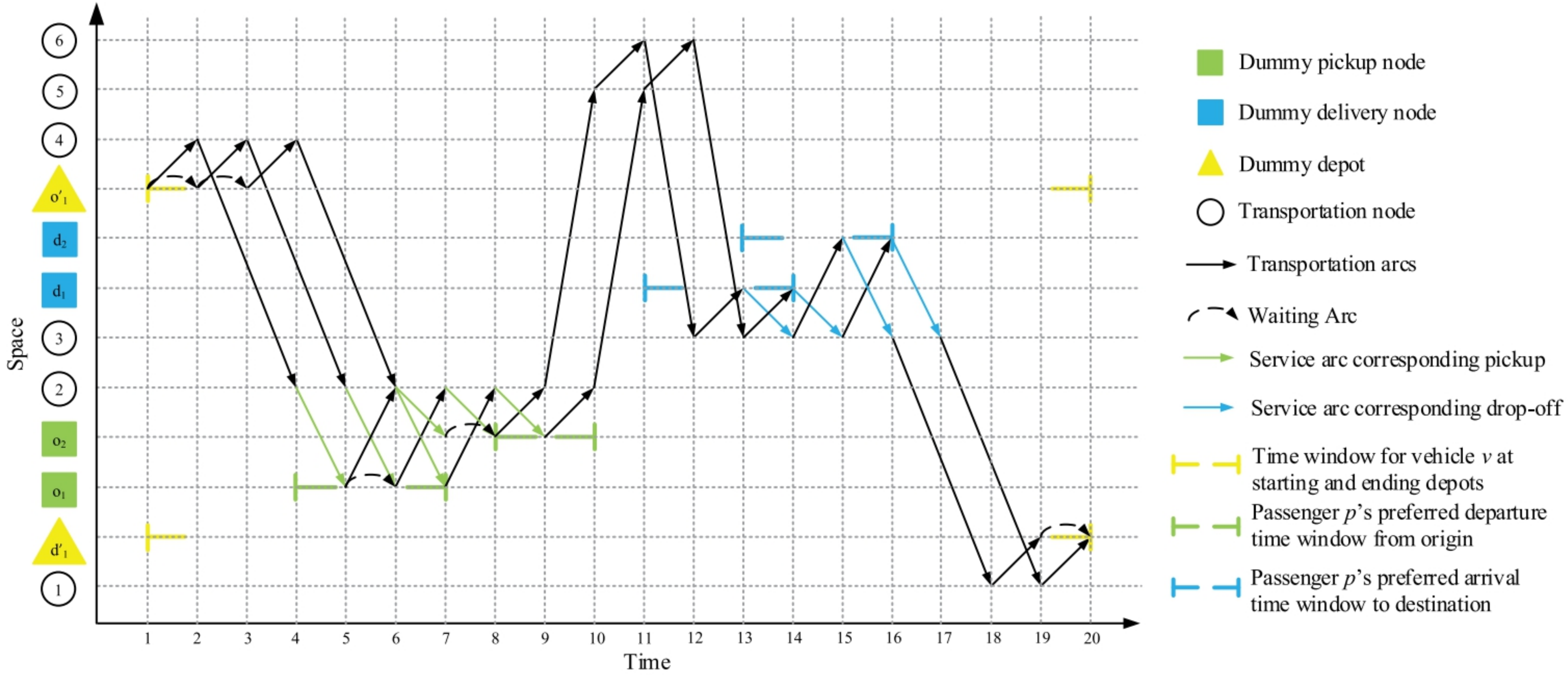
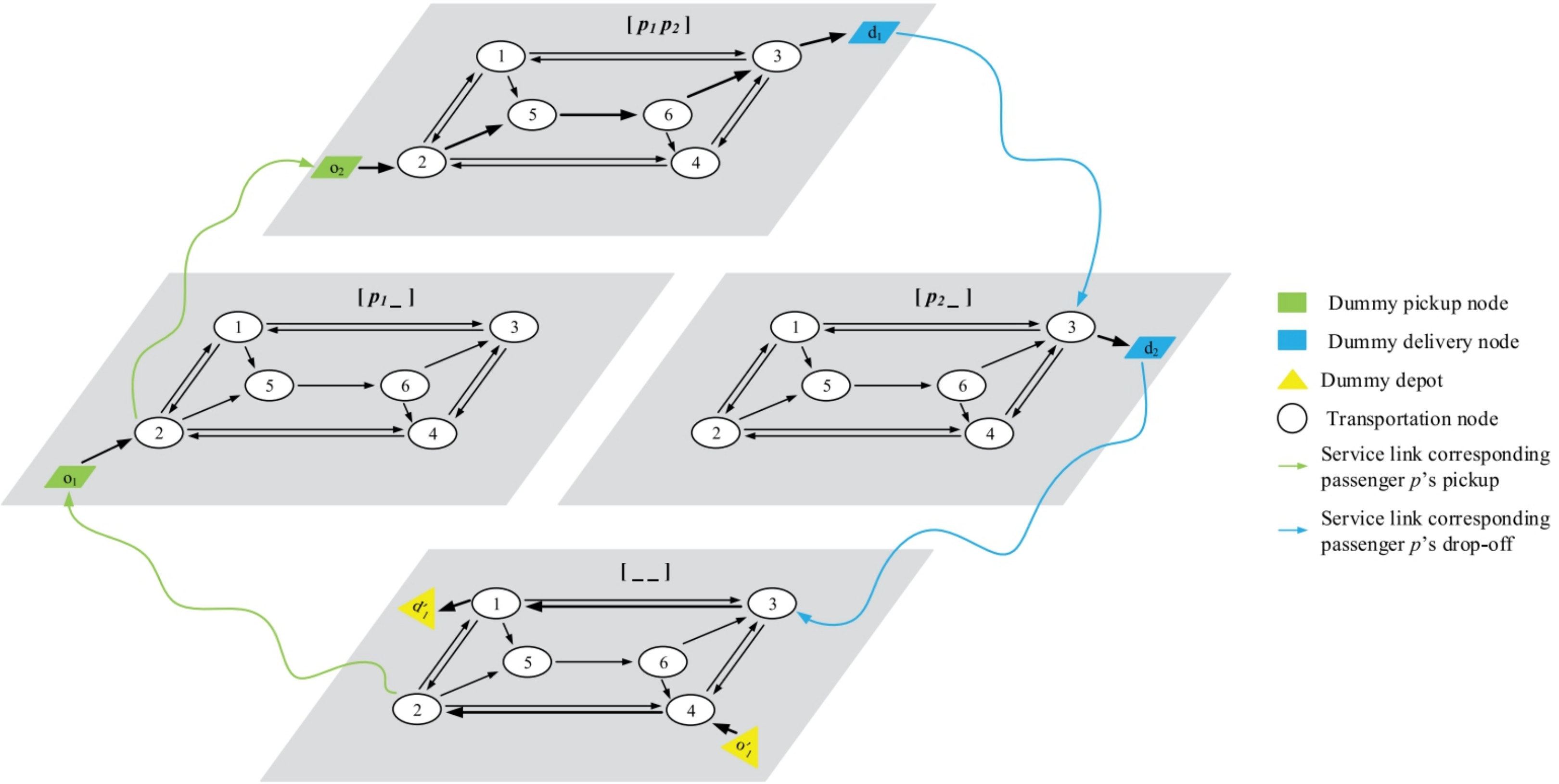



**Fig. 2.** Shortest paths with node sequence ($o'_1$, 4, 2, $o_1$, 2, $o_2$, 2, 5, 6, 3, $d_1$, 3, $d_2$, 3, 1, $d'_1$) in vehicle 1's space–time network.

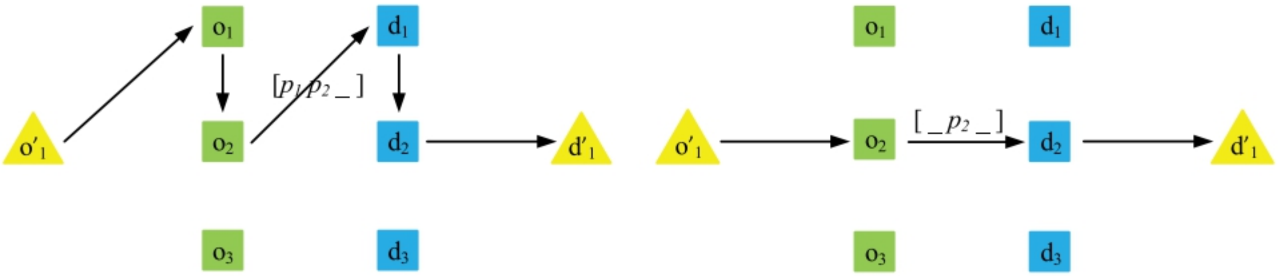
## Introduction

### State transition



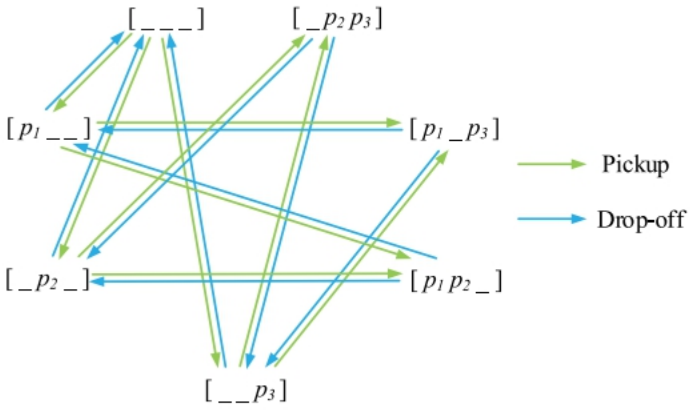
# Introduction

## State dimension




All possible combinations of passenger carrying states.

| $w/w'$ | [_ _ _] | [$p_1$ _ _] | [_ $p_2$ _] | [_ _ $p_3$] | [$p_1$ $p_2$ _] | [$p_1$ _ $p_3$] | [_ $p_2$ $p_3$] |
|---|---|---|---|---|---|---|---|
| [_ _ _] | no change | pickup | pickup | pickup | | | |
| [$p_1$ _ _] | drop-off | no change | | | pickup | pickup | |
| [_ $p_2$ _] | drop-off | | no change | | pickup | | pickup |
| [_ _ $p_3$] | drop-off | | | no change | | pickup | pickup |
| [$p_1$ $p_2$ _] | | drop-off | drop-off | | no change | | |
| [$p_1$ _ $p_3$] | | drop-off | | drop-off | | no change | |
| [_ $p_2$ $p_3$] | | | drop-off | drop-off | | | no change |



## Mathematical Model in the paper

Minimize total costs

$$\text{Min } Z = \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in B_v} c(v,i,j,t,s,\ w,w')y(v,i,j,t,s,w,w') \quad (1)$$

s.t.

Flow balance constraints at vehicle $v$'s origin vertex

$$\sum_{(i,j,t,s,\ w,w') \in B_v} y(v,i,j,t,s,\ w,w') = 1 \quad i = o'_v,\ t = e_v,\ w = w' = w_0,\ \forall v \in (V \cup V^*) \quad (2)$$

Flow balance constraint at vehicle $v$'s destination vertex

$$\sum_{(i,j,t,s,\ w,w') \in B_v} y(v,i,j,t,s,w,w') = 1 \quad j = d'_v, s = l_v,\ w = w' = w_0,\ \forall v \in (V \cup V^*) \quad (3)$$

Flow balance constraint at intermediate vertex

$$\sum_{(j,s,w'')} y(v,i,j,t,s,\ w,w'') - \sum_{(j',s',w')} y(v,j',i,s',t,\ w',w) = 0 \ (i,t,w) \notin \{(o'_v,e_v,\ w_0),(d'_v,l_v,\ w_0)\},\ \forall v \in (V \cup V^*) \quad (4)$$

Passenger $p$'s pickup request constraint

$$\sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Psi_{p,v}} y(v,i,j,t,s,w,w') = 1 \quad \forall p \in P \quad (5)$$

Variables denote the space-time-state trajectories of vehicles

Binary definitional constraint

$$y(v,i,j,t,s,w,w') \in \{0,\ 1\} \quad \forall (i,j,t,s,w,w') \in B_v, \quad \forall v \in (V \cup V^*) \quad (6)$$

## Lagrangian Relaxation Model

Lagarangian Relaxation

Lagarangian Multipliers

mean passengers’ profit

$$L = \sum_{v\in(V\cup V^*)} \sum_{(i,j,t,s,w,w')\in B_v} c\left(v,i,j,t,s,\ w,w'\right) y\left(v,i,j,t,s,w,w'\right) + \sum_{p\in P} \lambda(p) \left[ \sum_{v\in(V\cup V^*)} \sum_{(i,j,t,s,w,w')\in \Psi_{p,v}} y\left(v,i,j,t,s,w,w'\right) - 1 \right] \quad (7)$$

Therefore, the new relaxed problem can be written as follows:

$$\text{Min } L \quad (8)$$

s.t.

$$\sum_{(i,j,t,s,\ w,w')\in B_v} y\left(v,i,j,t,s,\ w,w'\right) = 1 \quad i = o'_v,\ \ t = e_v,\ \ w = w' = w_0,\ \ \forall v \in (V \cup V^*) \quad (9)$$

$$\sum_{(i,j,t,s,\ w,w')\in B_v} y\left(v,i,j,t,s,w,w'\right) = 1 \quad j = d'_v,\ s = l_v,\ \ w = w' = w_0,\ \ \forall v \in (V \cup V^*) \quad (10)$$

$$\sum_{(j,s,w'')} y\left(v,i,j,t,s,\ w,w''\right) - \sum_{(j',s',w')} y\left(v,j',i,s',t,\ w',w\right) = 0 \ \ (i,t,w) \notin \left\{\left(o'_v,e_v,\ w_0\right),\left(d'_v,l_v,\ w_0\right)\right\},\ \forall v \in (V \cup V^*) \quad (11)$$

$$y\left(v,i,j,t,s,w,w'\right) \in \{0,\ 1\} \quad \forall\left(i,j,t,s,w,w'\right) \in B_v,\quad \forall v \in (V \cup V^*) \quad (12)$$

If we further simplify function $L$, the problem will become a time-dependent least-cost path problem in the constructed state–space–time network. The simplified Lagrangian function $L$ can be written in the following form:

$$L = \sum_{v\in(V\cup V^*)} \sum_{(i,j,t,s,w,w')\in B_v} \xi\left(v,i,j,t,s,\ w,w'\right) y\left(v,i,j,t,s,w,w'\right) - \sum_{p\in P} \lambda(p) \quad (13)$$

**Input data:**

**Output data:**

Initial profit of passengers

**Input_node**

| node_id | node_typ | passenge | timestam | baseprof | x | y |
|---|---|---|---|---|---|---|
| 1001 | 1 | 1 | 32 | 20 | 61.671 | 71.431 |
| 2002 | 2 | 2 | 36 | | 62.22 | 69.113 |
| 2001 | 1 | 2 | 26 | 20 | 60.451 | 70.882 |
| 1002 | 2 | 1 | 41 | | 60.939 | 68.198 |
| 5 | | | | | 62.83 | 66.063 |
| 6 | | | | | 58.194 | 69.967 |
| 500 | | | | | | |
| 501 | | | | | | |
| 600 | | | | | | |

**Input_link**

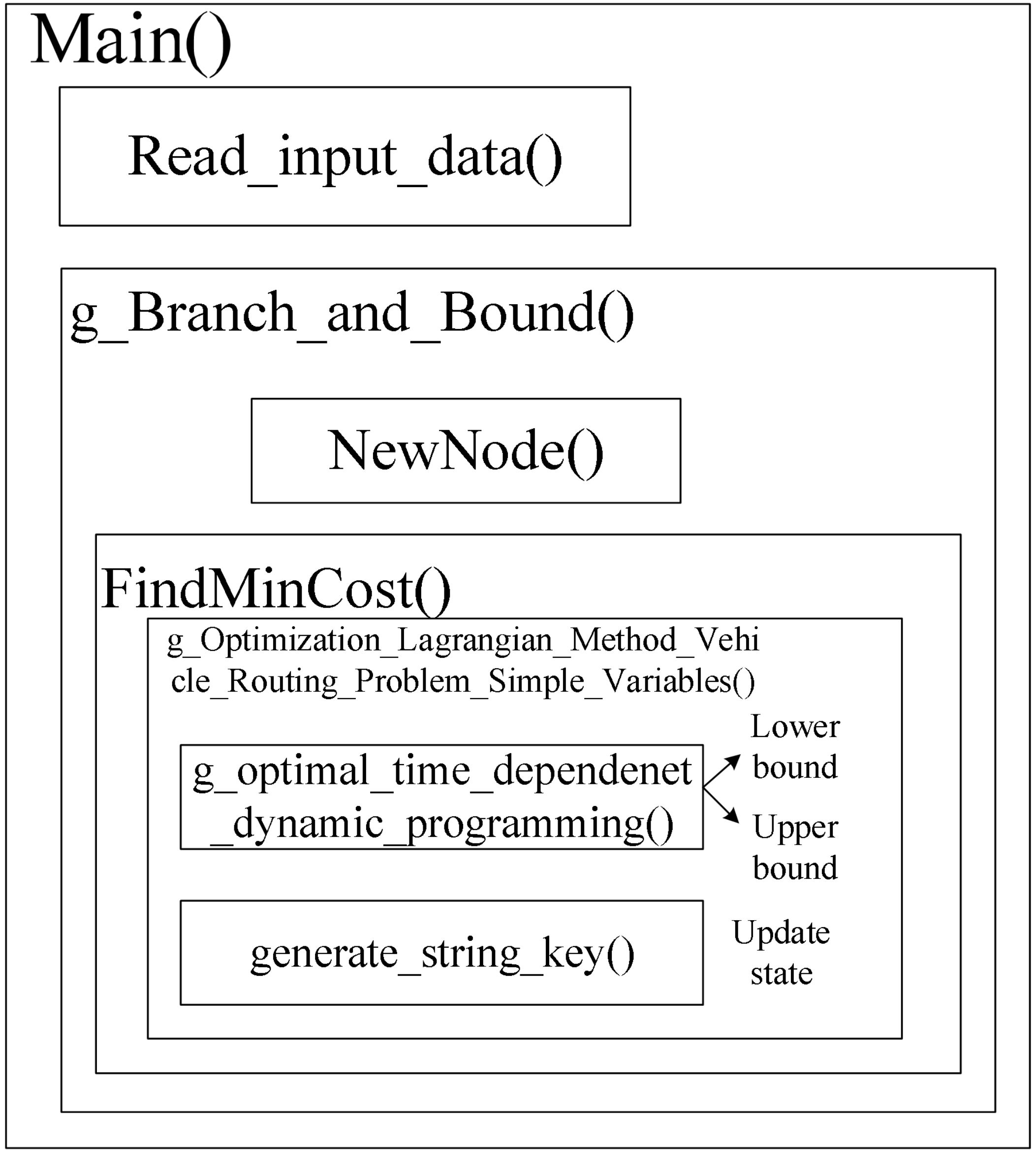
| from_node_id | to_node_id | link_type_name | travel_time |
|---|---|---|---|
| 1001 | 2002 | | 5 |
| 1001 | 2001 | | 5 |
| 1001 | 5 | | 3 |
| 2002 | 1 | | 5 |
| 2002 | 1002 | | 5 |
| 2002 | 5 | | 3 |

**Input_agent**

| agent_id | agent_type | from_node_id | to_node_id | departure_time | arrival_time | capacity |
|---|---|---|---|---|---|---|
| 1 | 1 | 501 | 600 | 0 | 130 | 3 |
| 2 | 1 | 502 | 600 | 0 | 130 | 3 |
| 3 | 1 | 503 | 600 | 0 | 130 | 3 |
| 4 | 1 | 504 | 600 | 0 | 130 | 3 |
| 5 | 1 | 505 | 600 | 0 | 130 | 3 |
| 6 | 1 | 506 | 600 | 0 | 130 | 3 |

1. Lagrangian lower bound and primal upper bound
2. Passenger and vehicles' assignments and time-space routes

## Main functions




Generate_string_key():

```
::string generate_string_key()

 //std::string string_key;
 stringstream s;

 s << "n";
 s << current_node_id;  // space key
 for (std::map<int, int>::iterator it = passenger_service_state.begin();
 {
     s << "_";

     s << it->first << "["<< it->second << "]";

 }
```

Such as: n4_1[1]_3[2]_4[2], means node 4 with pickup of p1, drop-off p3 and drop-off of p4

Including nodeID, passenger ID, and pickup or delivery state

Convenient for visiting in dynamic programming

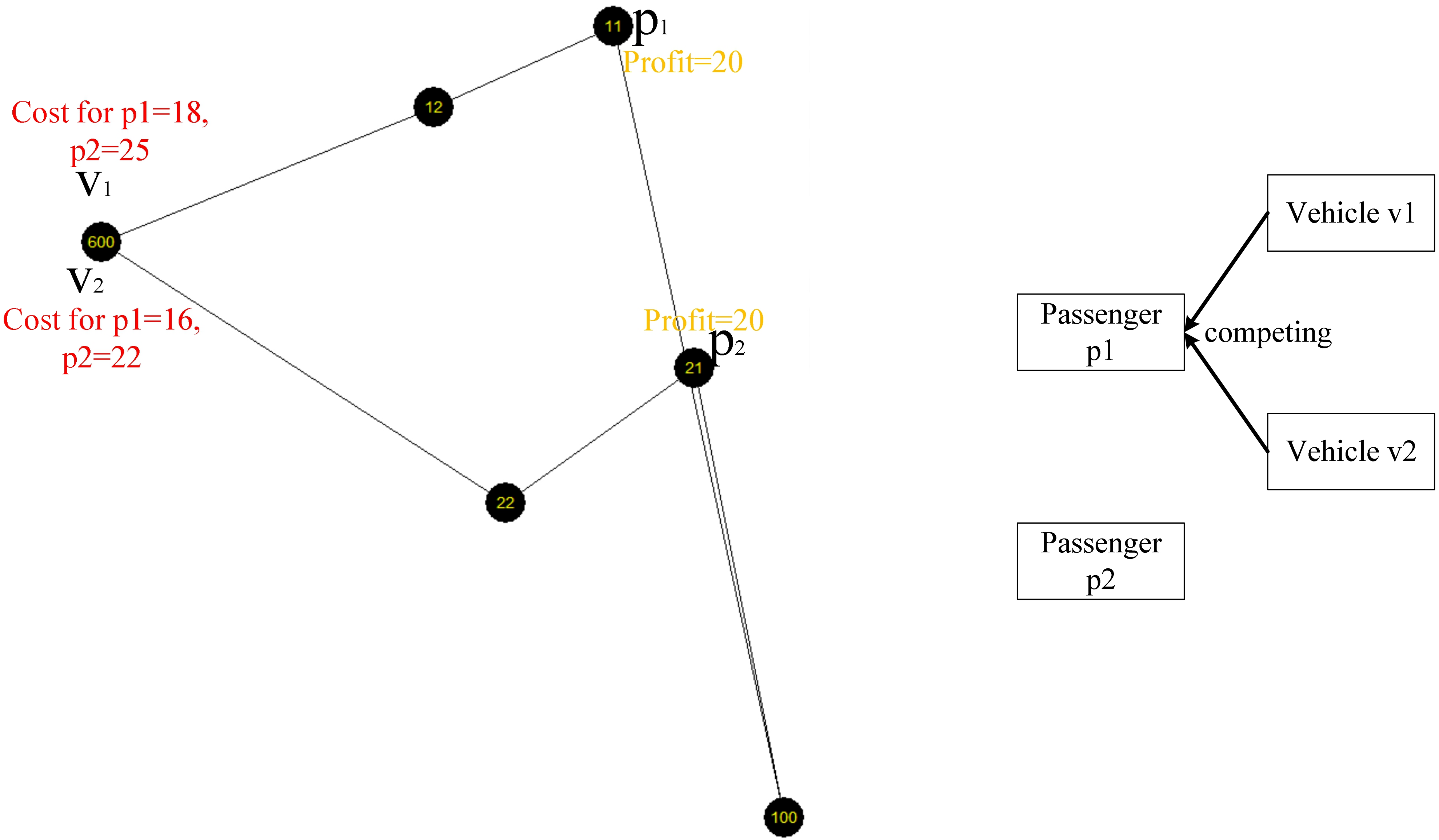
**Main Class**

| Class Name | Major functions | Major variables | Used modules |
|---|---|---|---|
| Node | CalculateCost() | PassengerID, VehicleID, LowerBound, UpperBound | Branch & Bound |
| VRP_exchange_data //in one node | AddP2VAssignment (int p, int v) | LowerBound, UpperBound, V2PAssignmentVector | Branch & Bound |
| V2PAssignment //vehicles assigned for each p | AddCompettingVehID (int vehicle_id) | input_prohibited_vehicle_id_vector: not allowable vehicles output_competting_vehicle_id_vector: competing vehicles | Branch & Bound |
| CVSState //for vehicle scheduling states | | LabelCost, PrimalLabelCost passenger_service_state | C_time_indexed_state_vector &DP |
| C_time_indexed_state_vector | update_state (CVSState new_element) | m_VSStateVector //vector of CVSState | DP |

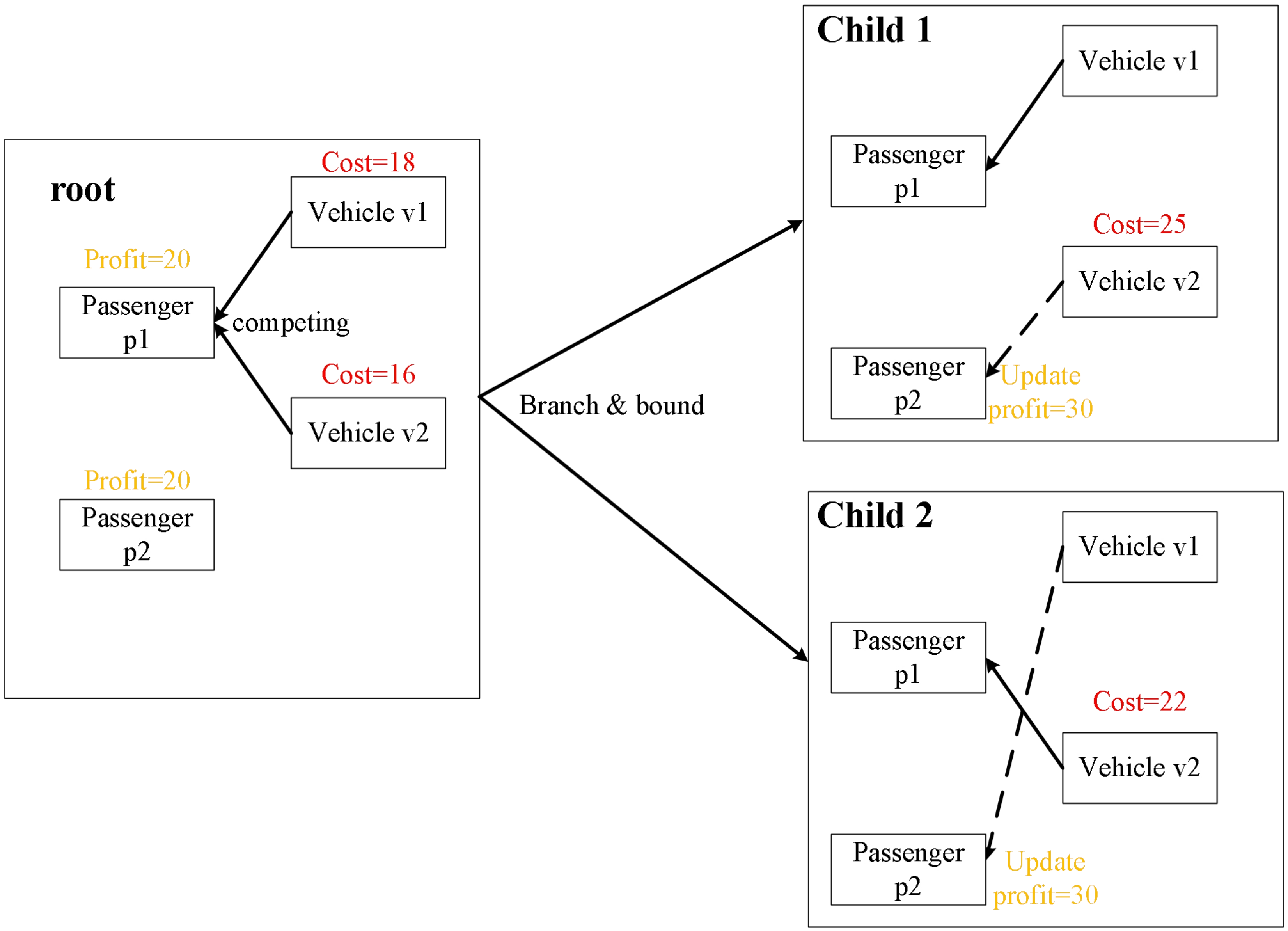
## Main variables

| For passengers | For vehicles |
|---|---|
| g_passenger_base_profit[passenger_id]<br>→profit of passengers (LR multipliers) | g_time_dependent_state_vector[vehicle_id][t].<br>m_VSStateVector<br>→state vector of a vehicle |
| competing_vehicle_size<br>→competing vehicels | g_ending_state_vector[vehicle_id]<br>→final state of a vehicle |
| input_prohibited_vehicle_id_vector<br>→in Branch and Bound, vehicles which cannot serve passenger p | |

# Branch & Bound



# Branch & Bound



# Branch & Bound

```
for (int p = 1; p < min_Node->l_vrp_data.V2PAssignmentVector.size(); p++) // for each p
{
    int competing_vehicle_size = min_Node->l_vrp_data.V2PAssignmentVector[p].output_competting_vehicle_id_vector.size();
    if (competing_vehicle_size >= 2) // if there are more than 2 competing vehicles
    {
        // first vi inclusive braches
        for (int vi = 0; vi < competing_vehicle_size; vi++)
        {
            BBDebugfile << "   Branch: pax " << p << " with competing v[ " << vi << "]=" << min_Node->l_vrp_data.V2PAssig

            // create a new tree node
            Node* child = newNode(min_Node->l_vrp_data, min_Node);
            int veh_id = min_Node->l_vrp_data.V2PAssignmentVector[p].output_competting_vehicle_id_vector[vi];
            child->VehicleID = veh_id;
            child->PassengerID = p;

            BBDebugfile << "   New Node " << child->node_id << " with vehicle " << child->VehicleID << " to Pax " << chil

            child->l_vrp_data.AddP2VAssignment(p, veh_id);  // designate p to veh_id
```

**[Assignment Problem Using Branch-And-Bound Method](https://math.stackexchange.com/questions/1466459/assignment-problem-using-branch-and-bound-method)**:
https://math.stackexchange.com/questions/1466459/assignment-problem-using-branch-and-bound-method

# Dynamic Programming

```
for (int t = departure_time; t <= arrival_time; t++)  //first loop: time
{
    // step 1: sort m_VSStateVector by labelCost for scan best k elements in step2 @CR
    g_time_dependent_state_vector[vehicle_id][t].Sort();

    // step 2: scan the best k elements
    for (int w_index = 0; w_index < min(BestKSize, g_time_dependent_state_vector[vehicle_id][t].m_VSStateVector.size()
    {
        CVSState* pElement = &(g_time_dependent_state_vector[vehicle_id][t].m_VSStateVector[w_index]);

        int from_node = pElement->current_node_id;
        // step 2.1 link from node to toNode
        for (int i = 0; i < g_outbound_node_size[from_node]; i++)
        {
            int to_node = g_outbound_node_id[from_node][i];
            int to_node_passenger_id = g_node_passenger_id[to_node];
            int to_node_type = g_node_type[to_node];
            int link_no = g_outbound_link_no[from_node][i];

            int next_time = max(g_node_timestamp[to_node], t + g_link_free_flow_travel_time[link_no]);
```

For vehicle v
  For time t
    for state w
      for outbound of current w's node i
      calculate labelCost of (i,t,w):
      if(the state does not exist)
        create it
      else if(the label cost of the temp state < label cost of the existing state)
        update the cost of the state

```
LabelCost -= g_passenger_base_profit[passenger_id];//LB
//LabelCost -= local_vehicle_passenger_additional_profit[vehicle_id][passenger_id];
LabelCost += max(0, passenger_service_time[passenger_id] - g_passenger_order_time[passenger_id]);
//UB--------------delay
PrimalLabelCost += max(0, passenger_service_time[passenger_id] - g_passenger_order_time[passenger_id]);
```

```
if (new_element.LabelCost < m_VSStateVector[state_index].LabelCost)
{
    m_VSStateVector[state_index].Copy(&new_element);
}
```

# Lagrangian Relaxation for Lower Bound

// step 1.2. update $LB^*$

– update $LB^k$ by substituting solution vector $Y^k_{LB}$ in the objective function of the dual problem (Ec

– update $LB^*$ by $max(LB^k, current\ LB^*)$ and $Y^*$ by its corresponding solution;

// step 1.3. sub-gradient calculation

– calculate the total number of visits of passenger $p$'s origin by expression (14);

$$\sum_{v\in(V\cup V^*)}\ \sum_{(i,j,t,s,w,w')\in\Psi_{p,v}} y(v,i,j,t,s,w,w') \tag{14}$$

– compute sub-gradients by Eq. (15);

$$\nabla L_{\lambda^k(p)} = \sum_{v\in(V\cup V^*)}\ \sum_{(v,i,j,t,s,w,w')\in\Psi_{p,v}} y(v,i,j,t,s,w,w') - 1 \text{ for } \quad \forall p \tag{15}$$

– update arc multipliers by Eq. (16);

$$\lambda^{k+1}(p) = \lambda^k(p) + \theta^k(p)\nabla L_{\lambda^k(p)} \text{ for } \quad \forall p \tag{16}$$

– update arc cost $\xi(v,i,j,t,s,\ w,w')$ for each arc $(v,i,j,t,s,w,w') \in \Psi_{p,v}$ by Eq. (17);

$$\xi(v,i,j,t,s,\ w,w') = \boxed{c(v,i,j,t,s,\ w,w')} + \boxed{\lambda^{k+1}(p)} \tag{17}$$

– update step size by Eq. (18);

$$\theta^{k+1}(p) = \frac{\theta^0(p)}{k+1} \tag{18}$$

```
LabelCost -= g_passenger_base_profit[passenger_id];//LB

LabelCost += max(0, passenger_service_time[passenger_id] - g_passenger_order_time[passenger_id]);

    LR_global_lower_bound += path_cost_by_vehicle_v;

    fprintf(g_pFileDebugLog, "LR_global_lower_bound += pat

}  //for each v
   //min CX + lamda(1 - # of visits)
for (int p = 1; p <= g_number_of_passengers; p++)
{
    LR_global_lower_bound += g_passenger_base_profit[p]
}
```

```
nt p = 1; p <= g_number_of_passengers; p++)

oat StepSize = 1 / (LR_iteration + 1.0f);
 (StepSize < g_minimum_subgradient_step_size)  //1.3.1 keep the minimum step size

  StepSize = g_minimum_subgradient_step_size;

t constant = 5;
passenger_base_profit[p] += constant*StepSize * (g_passenger_number_of_visits[p] - 1);
```

Sub-gradient algorithm

# Lagrangian Relaxation for Upper Bound

```
PrimalLabelCost += max(0, passenger_service_time[passenger_id] - g_passenger_order_time[passenger_id]);
```

```
LR_global_upper_bound += path_cost_by_vehicle_v;
```

Upper bound generates the solution without the profit of passengers.

# Example

| Node | Lower Bound | Upper Bound |
|---|---|---|
| 0 | 2 | 21 |
| 1 (v1-p1) | 10020 (cut) | |
| 2 (v2-p1) | 10020 (cut) | |
| 3 (p1 exclusive) | 10020 (cut) | |

```
number of nodes = 9
number of links = 19
read 9 nodes, 19 links, 2 passengers, 2vehicles

Computational time:,0
End of Lagrangian Iteration Process

Computational time:,0
End of Lagrangian Iteration Process

Computational time:,0
End of Lagrangian Iteration Process

Computational time:,0
End of Lagrangian Iteration Process
CPU Running Time = 27 milliseconds
End of Optimization
free memory..
```