|  |
| --- |
| **Guidelines for NeXTA&DTALite** |

|  |
| --- |
| Revision Date  2018-1-18 |

**CONTACT INFORMATION**

Further Details in https://github.com/xzhou99

For more information related to these Guidelines, please contact:

Xuesong Zhou ([xzhou74@asu.edu](mailto:xzhou74@asu.edu))

**TABLE OF CONTENTS**

[1. INTRODUCTION 1](#_Toc504051333)

[1.1 What is DTALite? 1](#_Toc504051334)

[1.2 System Design and Model Structure 1](#_Toc504051335)

[2. GETTING STARTED 3](#_Toc504051336)

[2.1 Installation Notes 3](#_Toc504051337)

[2.1.1. Hardware requirements 3](#_Toc504051338)

[2.1.2. DTALite/NeXTA Software Package 3](#_Toc504051339)

[2.2 Data Flow for Performing DTA 4](#_Toc504051340)

[2.3 Data Structure of Files 5](#_Toc504051341)

[2.3.1. Network Files 7](#_Toc504051342)

[2.3.2. Advanced Type Definition Files 12](#_Toc504051343)

[2.3.3. Simulation Configuration Files 15](#_Toc504051344)

[2.3.4. Scenario Files 29](#_Toc504051345)

[2.3.5. Output Files 31](#_Toc504051346)

[3. USER INTERFACES & BASIC CONTROLS 32](#_Toc504051347)

[3.1 User Interfaces 32](#_Toc504051348)

[3.1.1. GIS Interface 32](#_Toc504051349)

[3.1.2. Layer Control Panel 32](#_Toc504051350)

[3.1.3. Attribute Data Display Panel 33](#_Toc504051351)

[3.2 Management Toolbar 34](#_Toc504051352)

[3.2.1. Basic Management Tools 34](#_Toc504051353)

[3.2.2. Network Editing Tools 35](#_Toc504051354)

[3.2.3. Time Controlling Tools 36](#_Toc504051355)

[3.3 Viewing Modes 37](#_Toc504051356)

[3.3.1. Network View 37](#_Toc504051357)

[3.3.2. Animation View 37](#_Toc504051358)

[3.4 MOE Toolbar 38](#_Toc504051359)

[3.4.1. Density Visualization 38](#_Toc504051360)

[3.4.2. Volume Visualization 39](#_Toc504051361)

[3.4.3. Speed Visualization 40](#_Toc504051362)

[3.4.4. Queue Visualization 40](#_Toc504051363)

[3.4.5. Bottleneck Visualization 41](#_Toc504051364)

[3.4.6. Emissions Visualization 42](#_Toc504051365)

[3.5 Detailed Analytical Tools 42](#_Toc504051366)

[3.5.1. Link Analysis Tool 43](#_Toc504051367)

[3.5.2. Path Analysis Tool 43](#_Toc504051368)

[3.5.3. Vehicle Analysis Tool 44](#_Toc504051369)

[3.5.4. Summary Analysis Tool 47](#_Toc504051370)

[4. SCENARIO TESTING 52](#_Toc504051371)

[4.1. Basic Traffic Assignment Scenario 52](#_Toc504051372)

[4.2. Toll Scenario 63](#_Toc504051373)

[4.3. Emissions Scenario 66](#_Toc504051374)

[4.4. Work Zone Scenario 67](#_Toc504051375)

[4.5. VMS Scenario 72](#_Toc504051376)

[4.6. Signal Scenario (Link Based) 78](#_Toc504051377)

[4.7. Signal Scenario (Phase Based) 83](#_Toc504051378)

[4.8. Intermodal Scenario 100](#_Toc504051379)

[4.9. OD Matrix Estimation 103](#_Toc504051380)

**STEPS TO LEARNING**

|  |  |  |
| --- | --- | --- |
| **Step** | **Goal/Question** | **More Details in** |
| 1 | What is dynamic traffic assignment (DTA) ?  What is DTALite?  What is \*.tnp file? | Section 1  [1] DTALite A queue based mesoscopic traffic simulator for fast model evaluation and calibration |
| Install required software | Section 2.1 |
| 2 | Understand data flow for performing DTALite | Section 2.2 |
| 3 | Understand network files for DTALite input | Section 2.3.1 |
| Understand advanced Type Definition Files for DTALite input | Section 2.3.2 |
| Understand demand files required for DTALite input | Section 2.3.3 |
| Understand scenario files required for scenario settings | Section 2.3.4 |
| Understand output files from DTALite | Section 2.3.5 |
| 4 | Understand user interfaces and basic control in NeXTA | Section 3 |
| 5 | Learn how to import network form GIS \*.shp files in NeXTA | Section 4.1.1  [3] learning\_document\_GIS\_importing.docx |
| 6 | Understand different traffic flow thoery | [4] learning\_document\_equilibirum.docx  [5] learning\_document\_traffic\_flow\_theory.docx |
| 7 | Have a good understand in dynamic traffic assignment scenarios implemented in DTALite | Section 4.1  [6] learning\_document\_traveler\_information.docx |
| 8 | Understand what is OD matrix estimation (ODME) ?  Understand in ODME scenario implemented in DTALite | Section 4.2  [7] learning\_document\_OD Estimation.pdf” |
| 9 | Understand agent-based model | [8] learning\_document\_ABM DTA Day\_by\_day Integration.pdf”  [9] learning\_document\_DTA + ABM Integrated System .docx” |
| 10 | Case study | Dataset 1-9 in folder “DTALite-NEXTA-Software-Package” |

# INTRODUCTION

## What is DTALite?

Motivated by a wide range of application needs, such as region-wide traffic analysis and route guidance provision, dynamic traffic assignment (DTA) models have been increasingly recognized as an important tool for assessing operational performance of those applications at multiple spatial resolutions (e.g. network, corridor, and individual segment levels). The advances of DTA in this aspect are built upon the capabilities of DTA models in describing the formation, propagation, and dissipation of traffic congestion in a transportation network.

DTALite, an open-source mesoscopic DTA simulation package, in conjunction with the Network explorer for Traffic Analysis (NeXTA) graphic user interface, has been developed to provide transportation planners, engineers, and researchers with a theoretically rigorous and computationally efficient traffic network modeling tool. This fully functional, open-source DTA model can be downloaded from [https://github.com/xzhou99/dtalite\_software\_release](https://github.com/xzhou99/dtalite_software_release/tree/gh-pages/1_Software_Download)

In general, the software suite of DTALite + NeXTA aims to:

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

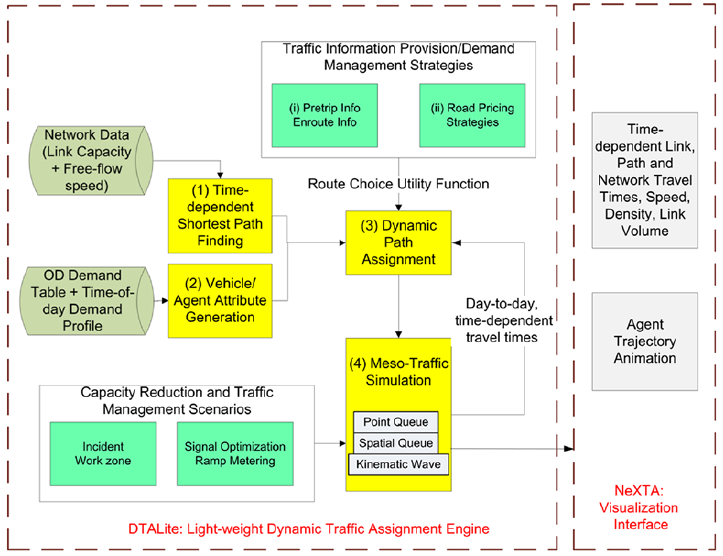
(2) Present results to other users by visualizing time-varying traffic flow dynamics and traveler route choice behavior in an integrated 2D/3D environment.

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

Additionally, DTALite also adopts a new software architecture and algorithm design to facilitate the most efficient use of emergent parallel (multi-core) processing techniques and exploit the unprecedented parallel computing power newly available on both laptops and desktops.

## System Design and Model Structure

The software architecture designed in DTALite aims to integrate many rich modeling and visualization capabilities into an open-source DTA model. Using a modularized design, the software suite of simulation engine and visualization interface can also serve future needs by enabling transportation researchers and software developers to continue to build upon and expand its range of capabilities. The streamlined data flow from static traffic assignment models and common signal data interfaces aims to allow planners and engineers to rapidly apply the advanced DTA methodology, and further examines the effectiveness of traffic mobility, reliability, and safety improvement strategies. The overall structure, illustrated in the figure blew, integrates the four major modeling components highlighted in yellow.



DTALite’s four major modeling components include:

(1) Time-dependent shortest path finding, based on a node-link network structure.

(2) Vehicle/agent attribute generation, which combines an origin–destination (OD) demand matrix with additional time-of-day departure time profile to generate trips.

(3) Dynamic path assignment module, which considers major factors affecting agents’ route choice or departure time choice behavior, such as (i) different types of traveler information supply strategies (e.g. historical, pre-trip, and/or en route information, and variable message signs) and (ii) road pricing strategies where economic values are converted to generalized travel time.

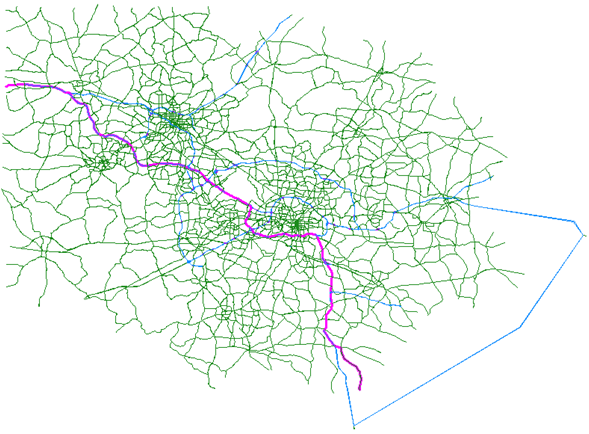
(4) A class of queue-based traffic flow models that can accept essential road capacity reduction or enhancement measures, such as work zones, incidents, and ramp meters.

# GETTING STARTED

## Installation Notes

### Hardware requirements

As shown in the figure below, a large-scale network provided by the NCSU team has about 2,300 Zones, 9,500 nodes, 20,000 links and 1,900 signals, and 1 million vehicles from 5AM to 10 AM, with 490K household.



The average running time is about 1 hours for 20 user equilibrium iterations on a PC with the following specification.

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

List of Internet Resources: https://github.com/xzhou99/learning-transportation (for hosting general learning material about network modelling)

### DTALite/NeXTA Software Package

#### Step 1: Download and unzip the NeXTA/DTALite software package

Please make sure you have unzipped the downloaded zip file “DTALite-NEXTA-Software-Package.zip”. The latest software release can be downloaded at [our Github website](https://github.com/xzhou99/dtalite_software_release) ( <https://github.com/xzhou99/dtalite_software_release>).

The software package includes two software applications: NEXTA as GUI and data hub; DTALite as DTA simulation engine. The DTALite and NeXTA package only run on Windows 7, Windows 10, Windows XP and Windows Vista.

#### Step 2: Install the Microsoft Visual C++ 2017 Redistributable Package (x64) and Gnuplot Software

Go to folder "Installation\_required\_software\_packages", install the Microsoft Visual C++ 2017 Redistributable Package (x64) <https://www.visualstudio.com/en-us/news/releasenotes/vs2017-relnotes> and Gnuplot Software for some visualization functions in NeXTA (<http://www.gnuplot.info/>).



#### Step 4: Click “NeXTA”--“File”--“Open Traffic Network Project” to choose the TNP file in your network data set

#### Step 5: Click “Project”--“Perform Traffic Assignment” to input your settings and run “DTALite” for dynamic traffic assignment and simulation

For importing the GIS information of traffic networks from external files, such as GIS shape files, please download the NeXTA\_GIS version from https://github.com/xzhou99/dtalite\_software\_release.

## Data Flow for Performing DTA

There are a number of practical challenges in enabling rapid data importing, analysis and exporting for multi-scale dynamic traffic analysis. The typical input data for mesoscopic DTA models include:

**Network Data**: Existing static traffic assignment network has a few link attributes but very different field naming conversions (e.g. from node, to node vs. A and B); the network topology has been widely coded in GIS format but with very different coordinate systems.

**Demand Data**: Trip tables at a TAZ level can be imported from regional traffic planning models for different peak and off peak periods, but the corresponding data files are available in many possible formats, such as column-by-column (origin, destination, value) vs. matrix, ASCII text.



The figure above gives the system importing, analysis and exporting data flow for the bundle of DTALite and NeXTA. In particular, our solution to import GIS data from multiple planning packages is to develop an open data format that allows DTALite users to feasibly convert their own data in proprietary format to a unifying data structure and widely used longitude and latitude coordinate system (WGS 84 used by Google Maps, Google Earth and on-line Google Fusion Tables), akin to the Open Document format which creates a standardized file format that allows users to open, edit, and save Microsoft Word documents from other applications such as Google Docs.

## Data Structure of Files

This section describes all input and output files associated with DTAlite/NeXTA package. All DTALite data files are in CSV format. Each input/output file includes descriptions for required variable names, followed by a short description of their type, purpose, function, interaction with other variables, and the use cases in which the variable is required/not required. Since NeXTA uses DTALite for transportation network analysis, not all variables required as inputs to DTALite are required as inputs for visualization in NeXTA, and not all variables required as inputs to NeXTA are required as inputs to DTALite.

Dynamic traffic assignment encompasses many problem formulation and representation methods, for example, trip-based vs. activity-based, fixed departure time vs. flexible departure time. As the DTA models expand to include greater behavioral and policy realism, the complexity of the model data structure increases. To reach the right balance between the representation details and required data preparation efforts, DTALite adopts an agent-based, simulation-based mesoscopic dynamic traffic assignment framework using time-dependent origin-destination matrices. Overall, it is a link-based simulation with capacity constraints. The capacity constraints make calibrating network capacities a very important step in the process of building our models. On the other hand, this modeling framework can be flexibly enhanced to meet specific modeling needs. For example, one advanced user can embed alternative traffic flow models to generate high-fidelity simulation results, or allow traveler-specific value of time attributes to recognize the heterogeneity of the network users in road pricing applications.

**Network data structure** defines the basic node-link structure used in DTALite and NeXTA, along with attributes for each link and node. Additionally, nodes are related to zones and activity locations, which can be used to disaggregate trips from zones to nodes and activity locations.

**Demand/agent data structure** describes the number of trips between zones or nodes. Through a demand meta data file, there are many different ways to define the time-dependent demand inputs: 1) Demand table with starting time and ending time, 2) Demand table with time-dependent 15-min departure time profile, and 3) Input vehicle file. Methods 1 and 2 will generate vehicles in the network based on the time period information provided. The vehicle data file describes all vehicle trips in the network, allowing the user to provide very detailed trip information, such as vehicle type, traveler information type, value of time, vehicle age as well as specific path sequences.

Below is a short list of key features for DTALite/NeXTA data files and data structure.

* Different layers: different files for node, link, zone, activity locations
* Many model attribute files: node control type, link type, demand type and vehicle type
* Time representation: 24 hour (for demand and sensor data), day number= iteration number, work zone has day attributes (for modelling day-to-day learning)
* Demand meta database file: “Dynamic demand data manager”, read multiple demand files, in different format: column, matrix, agent file, DYNASMART file, different demand loading periods, additional departure time profile
* Scenario setting file: traffic flow model, traffic assignment model, scenario number for multiple scenario runs
* Scenario files for advanced modelling features: work zone, incident, tolling, VMS files
* Sensor data file: for model validation and calibration, different time period
* Output files: simulation summary, network MOE, link MOE, trajectory file

|  |  |  |
| --- | --- | --- |
| **Data Block** | | **File Name** |
| Input Files | A: Network Files | A1: input\_node.csv |
| A2: input\_link.csv |
| A3: input\_zone.csv |
| A4: input\_activity\_location.csv |
| B: Advanced Definition Files | B1: input\_node\_control\_type.csv |
| B2: input\_link\_type.csv |
| B3: input\_demand\_type.csv |
| B4: optional\_vehicle\_type.csv |
| B5: optional\_vehicle\_emission\_rate.csv |
| C: Simulation Configuration Files | C1: input\_demand\_file\_list.csv |
| C2: input\_scenario\_settings.csv |
| C3: optional\_MOE\_settings.csv |
| D: Scenario Files | D1: Scenario\_Work\_Zone.csv |
| D2: Scenario\_Dynamic\_Message\_Sign.csv |
| D3: Scenario\_Link\_Based\_Toll.csv |
| Output Files | E: Simulation Output Files from DTALite | E1: [Output\_summary.csv](https://docs.google.com/spreadsheets/d/14P82EeRpBLBZT3VVKS1lY3MpcCFsCLu8ZW8dQAhGL4M/edit" \l "gid=2119997585) |
| E2: [Output\_NetworkTDMOE.csv](https://docs.google.com/spreadsheets/d/1shISC0ASaA784fFkqAmvWrEpE8G-56D30NTzKQdKDYE/edit" \l "gid=1865952138) |
| E3: [Output\_ODMOE.csv](https://docs.google.com/spreadsheets/d/12WRLPaVm5BfHp7xC3sbxoQTrttNNe1Pgu4foNmGeVB4/edit" \l "gid=1228094423) |
| E4: [Output\_linkMOE.csv](https://docs.google.com/spreadsheets/d/1S18Pf-D8F5fFAz7PaCK93nbhliTHw3SUcbX8a41L9qI/edit" \l "gid=1062729698) |
| E5: [Output\_linkTDMOE.csv](https://docs.google.com/spreadsheets/d/1gToOECJqT2T2rDpS6YSPRXZvz-xi4iN2iV32nrjK-AQ/edit" \l "gid=2142282830) |
| E6: [Output\_agent.csv](https://docs.google.com/spreadsheets/d/1Jrmq3tlV22qrwJb4tzHivMio04DNHZwHqdlJOcE_oIc/edit" \l "gid=689476001) |

### Network Files

High-level introductions:

* A generic network used for DTALite and NeXTA includes a set of four layers: node, link, activity location and zones.
* The specific file names are input\_node.csv, input\_link.csv, input\_zone.csv and input\_activity\_location.csv.
* A node has a certain control type such as pretimed signals or 4-way stop signs. A cycle length is required for signalized intersections/nodes.
* A link is defined using upstream node and downstream node ids, with essential attributes such as length, speed limit, number of lanes and capacity, typically required for static traffic assignment and mesoscopic traffic assignment.
* A zone can be a typical Traffic Analysis Zone (TAZ) that contains one or multiple centroids. The zone number is the index used in the OD demand table. The geometric field for a zone is not required for running DTA simulation, especially for external zones/stations.
* The activity location file serves as a mapping layer that maps zones to individual nodes. Activity locations are a special case of nodes for generating and attracting trip demand. In DTALite’s agent-based simulation method, an agent is a vehicle and it is generated based on an OD demand table from zone i to zone j with a certain vehicle type. An agent’s path is computed from one activity location in zone i to another activity location in zone j. One activity location can be a centroid (typically used in traditional static traffic assignment) or a physical node in the urban network.
* The node and link layers can use arbitrary coordinate system, but a WGS 84 (long/lat) coordinate system is preferred to export data to Google Earth/Google Map.
* The geometric attribute of node, link and zone layers is coded in a field of “geometry” for individual records. The “geometry” field uses the format required by Google Fusion Tables, so one can easily import node, link and zone data to Google Earth/Google Maps.
* NeXTA provides a number of methods to import network data from GIS planning packages in generic shape file format, Excel files, Openstreetmap OSM format and Synchro UTDF format.
* A user can also manually create a new network from the scratch by using a click-and-draw method based on a background image file.

#### input\_node.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| name | Optional for visualization only | Main street @ Highland Dr. |
| node\_id | Node identification number | 1001 |
| control\_type | Intersection control type, defined in input\_node\_control\_type.csv. Typical range: 1-7. E.g. 5: pretimed signal. | 5 |
| control\_type\_name | The text name corresponding to the control type number in the control\_type field. This text is automatically generated based on the numeric value in field control\_type when a user saves the network in NeXTA. | input pretimed\_signal |
| cycle\_length\_in\_second | The signal cycle length for a specific node, unit: second. | 120 |
| x | Longitude or horizontal coordinate in any arbitrary geographic coordinate system. | 100 |
| y | Latitude or vertical coordinate horizontal coordinate in any arbitrary geographic coordinate system | 200 |
| geometry | Text string used to describe node location (typically in WGS84 geographic coordinate system) | <Point><coordinates>-111.979958,40.703899</coordinates></Point> |

**Remarks:**

1. **Minimum set of fields:** node\_id, x and y. The geometry field can be generated from NeXTA based on fields x and y. The control\_type (specifically signal location) can be identified by NeXTA infile.

#### input\_link.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| name | Optional for visualization purposes | Main Street |
| link\_id | Optional link identification number, generated from NeXTA | 101 |
| from\_node\_id | Upstream node number of the link, must already defined in input\_node.csv | 2 |
| to\_node\_id | Downstream node number of the link, must already defined in input\_node.csv | 3 |
| link\_type\_name | Optional text label for visualization and data checking purposes, generated from NeXTA based on field link\_type and name defined in input\_link\_type.csv | Minor arterial |
| direction | Identifies the direction of travel on the link. Default value = 1. When -1, NeXTA reverses from\_node\_id and to\_node\_id for correct orientation. When 0 or 2, NeXTA automatically converts link into two one-way links. | 1 |
| length | The length of the link (between end nodes), measured in units of miles. | 1.0 |
| number\_of\_lanes | The number of lanes on the link | 2 |
| speed\_limit | Speed limit on defined link in units of miles per hour. Unit: mph | 20 |
| lane\_cap | Maximum service flow rate for each lane on the link, in vehicles per hour. This capacity value is not used used for signalized control with a positive cycle length defined. Typical values are 1800 for a freeway lane, 900 for an arterial lane, 1500 for a single on-ramp lane. | 1500 |
| link\_type | Link type identification number, corresponding to link functional class (freeway, ramps), must be defined in input\_link\_type.csv | 7 |
| traffic\_flow\_model | Traffic flow model can be defined for each link respectively. 0 for BPR function while 1 for point queue model. | 0 |
| jam\_density | Jam density (in vehicles per mile per lane), input for two traffic flow models (spatial queue and Newell’s simplified kinematic wave model). Unit: number of vehicles per mile per lane: default value 180 for freeway links, 190 for arterial street links. | 180 |
| wave\_speed | Backward wave speed in miles per hour, only used in Newell’s simplified kinematic wave model to define the vehicle storage space on a link: Default = 12 mph | 12 |
| geometry | Text string used to describe link shape and location (typically in WGS84 geographic coordinate system). The initial value can be empty, and NeXTA will generate the text string based on the coordinates of upstream and downstream nodes. | <LineString><coordinates>4165.673828,23656.343750,0.0 5207.092773,23656.343750,0.0</coordinates></LineString> |

**Remarks:**

1. **Requirements:** the from\_node\_id, to\_node\_id fields must be defined in input\_node.csv. If the downstream node is a signalized intersection, the effective\_green\_time\_length\_in\_second field is required.
2. **Minimum required set of fields:** from\_node\_id, to\_node\_id, length\_in\_mile, number\_of\_lanes, speed\_limit\_in\_mph, lane\_capacity\_in\_vhc\_per\_hour, link\_type.
3. **Fields can be generated or populated by NeXTA:**

geometry fields can be imported from GIS shape files or generated based on the coordinates of upstream and downstream nodes. direction = 1 by default. saturation\_flow\_rate\_in\_vhc\_per\_hour\_per\_lane =2000 by default. jam\_density\_in\_vhc\_pmpl is 180 by default for freeway links, 190 for arterial links. wave\_speed\_in\_mph is 12 by default.

effective\_green\_time\_length\_in\_second can be computed as cycle length\*lane\_capacity/saturation flow rate. E.g. 120 second cycle length \* 900 lane capacity / 1800 saturation flow rate = 60 as effective green time.

#### input\_zone.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| zone\_id | Zone identification number. A zone number used in OD demand table files must be defined here first. Zone numbers are not required to be consecutively sequential. | 1001 |
| geometry | Optional: Text string used to describe zone location for visualization (typically in WGS84 geographic coordinate system). | “<Polygon><outerBoundaryIs><LinearRing><coordinates>-111.907241,40.802401,0.0 -111.870550,40.789266,0.0 -111.822437,40.768850,0.0 -111.907241,40.802401,0.0<coordinates> </LinearRing></outerBoundaryIs></Polygon>" |

#### input\_activity\_location.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| zone\_id | Zone identification number | 1001 |
| node\_id | Node identification number associated with the zone identification number in the same row. Must defined in input\_node.csv. | 1001 |
| external\_OD\_flag | Used to identify the type of activity location as non-external (default = 0) or external (-1 or 1). When 0, acts as both origin and destination. | 0 |

**Remarks:**

1. **Requirements:** A zone used in OD demand tables (with a positive number of trips generated) must contain at least one node as it activity locations. node\_id must be defined in input\_node.csv.

### Advanced Type Definition Files

High-level introductions:

* DTALite and NeXTA aim to accommodate and preserve flexible data types defined by users. We do not impose limitations on the number of link types, demand types, and users can use their existing link type, demand type as is to maintain a seamless connection from their existing static planning models to the new DTA tools.
* Typical definitional files include input\_node\_control\_type.csv, input\_link\_type.csv, input\_demand\_type.csv.
* When creating a new data set, a user does not need to create those files manually and NeXTA can fetch all default files to the current project folder automatically.
* A number of mapping fields are provided to relate the user-defined type to the essential data type used in DTALite. E.g. link type 100 is mapped to the freeway type in DTALite through a code of f . A demand type of 3 is mapped to the truck type in DTALite as pricing type of 2.
* DTALite performs an agent-based routing for road pricing applications. with the following individual generalized cost function. 

#### input\_node\_control\_type.csv

The input\_node\_control\_type table defines the control type of nodes in the network in terms of control type name, unknown control, no control, yield sign, 2way stop sign, 4way stop sign, pretimed signal, actuated signal and roundabout. This file is required when using the network import tool, and the control type field is read from the node shape file.

#### input\_link\_type.csv

The input\_link\_type table allows users to define their own specific link types, as long as the flag variables are correctly used to identify how the different link types are connected/related (e.g., freeways connect to arterials using ramps). Only one flag may be used for each link type. Link types can also be used to determine how links are visualized in NeXTA. This file is required when using the network import tool to interpret the link type field in the link shape file.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| link\_type | Link type identification number | 100 |
| link\_type\_name | Optional: Name label assigned to link type in the same row, used for visualization purposes in NeXTA | freeway |
| type\_code | A text character which identifies which type of link is mapped to the link type identification number. f = freeway, h = highway/expressway, a = arterial, c = connector, r = ramp, t = transit, w = walk | f |
| default\_lane\_capacity | According to the link type, the lane capacity assigned by default to new links created in NeXTA. | 1900 |
| default\_speed\_limit | According to the link type,the speed limit assigned by default to new links created in NeXTA | 60 |
| default\_number\_of\_lanes | According to the link type, the new of lanes assigned by default to new links created in NeXTA. | 3 |

#### input\_demand\_type.csv

The input\_demand\_type table is used to define the characteristics for different demand types for the trips in demand files such as input\_demand.csv. There are three different demand types by default (1 = SOV, 2 = HOV, 3 = Trucks), but additional types can be defined in the table (e.g., trip purpose – HBW, HBO, etc.).

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| demand\_type | Demand type identification number | 1 |
| demand\_type\_name | Optional: Name label assigned to demand type in the same row, used for visualization purposes in NeXTA | SOV |
| avg\_VOT | Average Value of Time (in units of dollars/hour) assigned to the demand type in the same row. This value is not used when the file input\_VOT.csv is provided, which contains more detailed distributions for VOT. | 10 |
| percentage\_of\_pretrip\_info | Percentage of vehicles with pre-trip travel time information. Affects routing behavior in DTALite. | 5 |
| percentage\_of\_enroute\_info | Percentage of vehicles with en-route travel time information. Affects routing behavior in DTALite. Drivers with historical information = 100 – percentage\_of\_pretrip\_info – percentage\_of\_enroute\_info. | 5 |
| percentage\_of\_vehicle\_type1 | Percentage of vehicles of vehicle type 1 for the demand type in the same row. Percentages in row should sum to 100. | 80 |
| percentage\_of\_vehicle\_type2 | Percentage of vehicles of vehicle type 2 for the demand type in the same row. Percentages in row should sum to 100. | 20 |
| percentage\_of\_vehicle\_type# | Additional columns (with incremental #) can be used when more vehicle types are defined. | 0 |

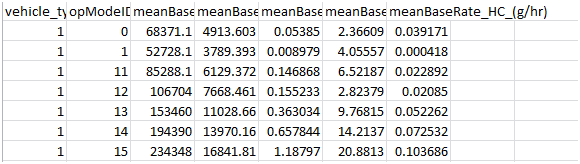
#### optional\_vehicle\_type.csv

The input\_vehicle\_type table is used to define different vehicle types for emissions analysis.

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| vehicle\_type | Vehicle type identification number |
| vehicle\_type\_name | Name label assigned to vehicle type in the same row |
| rolling\_term\_a  rotating\_term\_b  drag\_term\_c  source\_mass | Fields used for calculating vehicle-specific power (VSP) |
| percentage\_of\_age\_0  percentage\_of\_age\_5  percentage\_of\_age\_10  percentage\_of\_age\_15 | Percentage of vehicles in specific age group |

#### optional\_vehicle\_emission\_rate.csv

It should be kept with the default values. A sample file is shown in the figure below.



### Simulation Configuration Files

High-level introductions:

* DTALite and NeXTA aim to accommodate and preserve flexible demand file format used by users. DTALite can use the demand file as is.
* DTALite accepts multiple demand file types, such as 3-column, NXN matrix, agent-base file, DYNASMART format.
* Through a very flexible demand meta database configuration file, a user only needs to provide specific information to map demand type and demand loading time periods from their existing demand table to the new DTA tools.
* To create time-dependent demand loading patterns, a user can define loading ratio for each 15 min interval.
* DTALite uses a 24-hour demand horizon /representation to facilitate 24-hour dynamic traffic assignment and compare simulated results with sensor data easily (defined in the format of 24-hour horizon).
* Typical configuration files include meta demand database, scenario setting, MOE settings.

#### input\_demand\_file\_list.csv

The input\_demand\_file\_list table is used to define the characteristics of demand data.

Through a temporal demand profile table per record, users can define the proportion of demand in the network as a function of time, which is used to initiate trips in the simulation over the modeling horizon. This table can be used to supplement demand type information in an input demand table, where DTALite will use the temporal demand profile information in place of other time information.

This meta-data file requires several entries, but the relevant entries are as explained below:

1. Specify the demand file name (e.g., sov\_14\_15.mtx) and format (e.g., column)
2. Specify the number of lines in the demand file to be skipped by NeXTA (8 for MTX file)
3. Indicate whether subtotals are present in the last column (zero for none)
4. Specify the loading start time and end time for the demand file (840 to 900, or 2PM to 3PM)
5. Specify the demand types associated with the demand file (only demand\_type1)

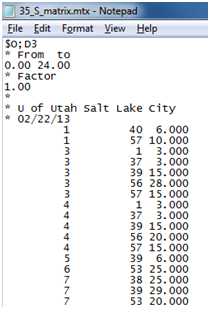
|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Acceptable Values** | **Description** | **Sample Value** |
| scenario\_no | Value ≥ 0 | Scenario identification number | 1 |
| file\_sequence\_no | Value > 0 | File identification number | 1 |
| file\_name | demand.dat | Name of demand file |  |
| format\_type | column,matrix, full\_matrix, agent\_csv, agent\_bin, trip\_csv, transims\_trip\_file | Input file format type |  |
| number\_of\_lines\_  to\_be\_skipped | Value ≥ 0 | The number of lines to be skipped at the beginning of demand file | 0 |
| loading\_multiplier | Value > 0 | Local multiplication factor applied to the number of trips in the demand file |  |
| start\_time\_in\_min | 0 to 1440 | Demand loading start time, which is the time gap in min from 0:00 |  |
| end\_time\_in\_min | 0 to 1440 | Demand loading end time, which is the time gap in min from 0:00 |  |
| apply\_additional\_  time\_dependent\_  profile | 0 or 1 | 0: not use the time dependent profile in this table, that is, a flat demand pattern is used between time period [start\_time\_in\_min, end\_time\_in\_min]  1: use the time dependent profile in this table |  |
| subtotal\_in\_last\_  column | 0 or 1 | flag used for subtotal in last column of matrix demand file |  |
| number\_of\_  demand\_types | Value ≥ 1 | Number of demand types stored in demand file |  |
| demand\_type\_in\_3rd\_column | default value 0  1: a demand type is specified for each record | demand type for format:  origin, destination, demand value, value | 2,100,1, 10  2,100,2, 20  zone 2 to zone 100, 10 vehicles for demand type 1  zone 2 to zone 100, 20 vehicles for demand type 2 |
| demand\_type\_1 | Value ≥ 1 |  |  |
| demand\_type\_2 | Value ≥ 1 |  |  |
| demand\_type\_3 | Value ≥ 1 |  |  |
| demand\_type\_4 | Value ≥ 1 |  |  |
| '00:00 | 0 to 1 | Proportion of demand in specified time interval compared to 24-hour time period |  |
| '00:15 | 0 to 1 | Proportion of demand in specified time interval compared to 24-hour time period |  |
| … |  |  |  |
| '23:45 | 0 to 1 | Proportion of demand in specified time interval compared to 24-hour time period |  |

**Remarks:**

1. We first show a few examples to show how to configure the following key parameters for different demand files.
2. **format\_type:** subtotal\_in\_last\_column; number\_of\_demand\_types

#### Example 1: column format

Below is an OD demand file titled “35\_S\_matrix.mtx” from VISUM for both LOV and HOV demand types during 6AM and 11AM. This combined demand matrix has 90% LOV and 10% HOV.



The corresponding meta data configuration is shown at the following table. The main data block has a three-column format, and we need to skip the first 8 lines of comments. The first line after the comments reads combined demand 6.0 from zone 1 to zone 40.

As there are two demand types, we use two records with the same file name of “35\_S\_matrix.mtx”, but with different loading\_multiplier = 0.9 and 0.1 for different demand\_type\_1 = 1 (LOV for the first record) and demand\_type\_1 = 2 (HOV for the second record). The start\_time\_in\_min = 360 corresponds to 6AM and the end time of 540 refers to 9AM.

Demand\_type\_x means the x th field after the row and column indices in the three-column or multi-column format. DTAlite will have 0.9\*6.0 = 5.4 LOV vehicles, and 0.1\*6.0 for 0.6 HOV vehicles from zone 1 to zone 2. The 5.4 vehicles will be randomly rounded up or down according to a uniform distribution to convert to an integer number (5 or 6 vehicles) in the final simulation process.



#### Example 2.1: matrix format

Below is an OD demand matrix file titled “demand\_matrix.csv” for 4 zones (1,2,3 and 40), and the demand horizon is 7AM to 9AM. A users wants to apply time-dependent departure time profile with a relatively lower demand volume between 7AM and 8AM.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 40 | TOT |
| 1 | 0 | 1000 | 10 | 10 | 1020 |
| 2 | 10 | 0 | 10 | 10 | 30 |
| 3 | 10 | 10 | 0 | 10 | 30 |
| 4 | 10 | 10 | 10 | 0 | 30 |
| TOT | 30 | 1020 | 30 | 30 | 1110 |

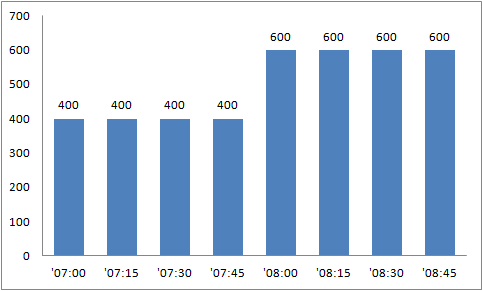
The corresponding input\_demand\_file\_list.csv is shown below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **file\_name** | **format\_type** | **number\_of\_lines\_to\_be\_skipped** | **loading\_multiplier** | **start\_time\_in\_min** | **end\_time\_in\_min** | **apply\_additional\_time\_dependent\_profile** | **subtotal\_in\_last\_column** | **number\_of\_demand\_types** | **demand\_type\_1** |
| demand\_matrix.csv | matrix | 0 | 2 | 420 | 540 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **'07:00** | **'07:15** | **'07:30** | **'07:45** | **'08:00** | **'08:15** | **'08:30** | **'08:45** |
| 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |

**Remark:**

1. file\_name: “demand\_matrix.csv” is the OD table for the AM period being brought into DTALite/NeXTA.
2. format\_type: “matrix” is the type of OD table format for this small test application. DTALite will skip the text string ‘TOT’ as it reads only numerical values.
3. Start\_time is minute 420 or 7 AM and the End\_time is minute 540 or 9 AM. The OD table stores number of trips between 7 and 9 AM. loading\_multiplier is 2, so a multiplier of 2 is applied to each row. That is, the total demand from zone 1 to zone 2 is 1000\*2 =2000.
4. subtotal\_in\_last\_column = 1 as there an origin-based summation for each zone in the matrix.
5. As field apply\_additional\_time\_dependent\_profile is set to 1, then a time-dependent departure time profile will be read and result in the following demand loading pattern. That is, the demand from 7:00AM to 7:15 AM from zone 1 to zone 2 has a demand volume of 1000\*2\*0.2 = 400. Those 400 vehicles will be uniformly assigned a departure time between 7:00AM to 7:15 AM, which leads to an average departure time interval of 15 min /401 intervals = 2.24 seconds. As a result, the first three vehicles depart from zone at 7:00:00, 7:00:02, and 7:00:04, respectively.



#### Example 2.2: full matrix format

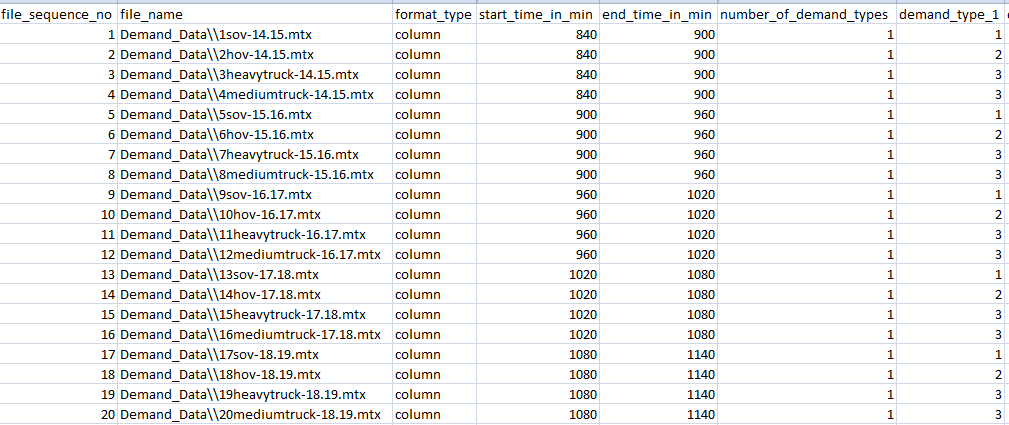
Below is an OD demand matrix file titled “demand\_matrix.csv” for 4 valid zones (1,2,3 and 10). Note that, zones 4,5,6..,9 are not defined in input\_zone.csv. A full matrix with sequential consecutive zone numbers (starting from zone 1 to the largest zone number 10) are listed for the first column and the first row, and all the OD pairs with invalid zone numbers (e.g. 1 to 4, 5, 6, 7,8,9; or zone 4 to all the other zones) have only values of zero.

|  |
| --- |
| trip 1 2 3 4 5 6 7 8 9 10  1 100 20 40 0 0 0 0 0 0 20  2 20 100 40 0 0 0 0 0 0 40  3 40 20 100 0 0 0 0 0 0 40  4 0 0 0 0 0 0 0 0 0 0  5 0 0 0 0 0 0 0 0 0 0  6 0 0 0 0 0 0 0 0 0 0  7 0 0 0 0 0 0 0 0 0 0  8 0 0 0 0 0 0 0 0 0 0  9 0 0 0 0 0 0 0 0 0 0  10 20 40 20 0 0 0 0 0 0 100 |

The corresponding input\_demand\_file\_list.csv is shown below, assuming the demand horizon is 7AM to 9AM.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **file\_name** | **format\_type** | **number\_of\_lines\_to\_be\_skipped** | **loading\_multiplier** | **start\_time\_in\_min** | **end\_time\_in\_min** | **apply\_additional\_time\_dependent\_profile** | **subtotal\_in\_last\_column** | **number\_of\_demand\_types** | **demand\_type\_1** |
| demand\_matrix.csv | full\_matrix | 0 | 2 | 420 | 540 | 0 | 0 | 1 | 1 |

#### Example 3: Multiple demand files for different demand types and hours



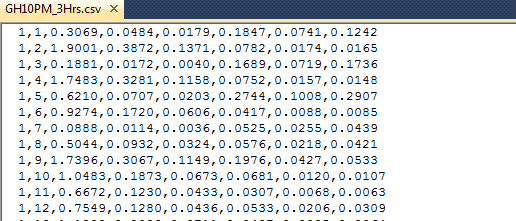
The screenshot above shows an example of working with existing demand files.

There are 20 hourly demand files exported from VISUM, and all those MTX files are put under a subfolder of Demand\_Data. There are 20 records in the above input\_demand\_file\_list.csv, and each record aims to map the data content to the overall time-dependent demand matrices used in DTALite. The first record specifies the following information:

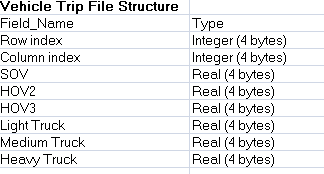
1. the demand file name (e.g., sov\_14\_15.mtx)
2. format (e.g., column)
3. the loading start time and end time for the demand file (e.g. 840 to 900, or 2PM to 3PM)
4. the demand types associated with the demand file (only demand\_type1), and demand types 1, 2 and 3 are SOV, HOV and truck, respectively.

#### Example 4: Multiple demand types as different columns per record from a TransCAD data set

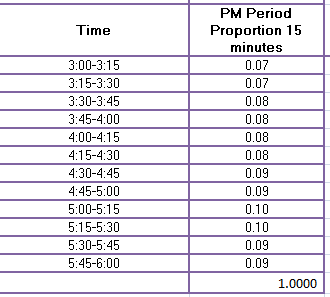
A TransCAD user has the following demand file titled GH10PM\_3Hrs.csv.



Each record includes demand values for 6 demand types, shown in the following data structure description.



They want to use the following departure time profile from 3PM to 6PM.



The corresponding input\_demand\_file\_list.csv is configured below.

The first part of the file is shown below, which indicates we use a “column” format, with demand loading period from 3PM (900 min) to 6PM (1089 min) with additional time-dependent demand profile (= 1).



The second part of file specifies 6 demand types to be loaded.

number\_of\_demand\_types = 6 for 6 demand types.

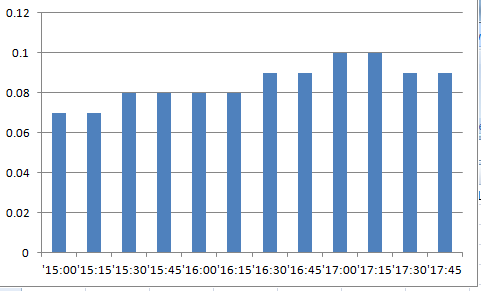
Field names demand\_type\_1, demand\_type\_2, demand\_type\_3, …, demand\_type\_6 requires demand type for each column after the row index and column index in the original demand file.

demand\_type\_1 = 1 for SOV type, demand\_type\_2 = for HOV2 and demand\_type\_3 = for HOV3. demand\_type\_4= demand\_type\_5 = demand\_type\_6 = 3 for different types of trucks.



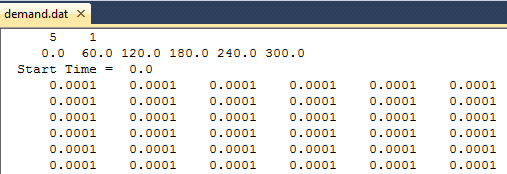
The departure time profile is specified in the third part of the meta data file. Field ‘15:00 has a value of 0.07 for the time period between 15:00 and 15:15.

 leads to the following demand loading pattern.



#### Example 5: Multiple time-dependent demand files from a DYNASMART data set

A typical DYNASMART demand data includes three file: demand.dat, demand\_HOV.dat and demand\_truck.dat. The following screenshot shows a demand.data file for a 6-hour horizon from 6AM to 11AM.



We can construct the following input\_demand\_file\_list.csv.

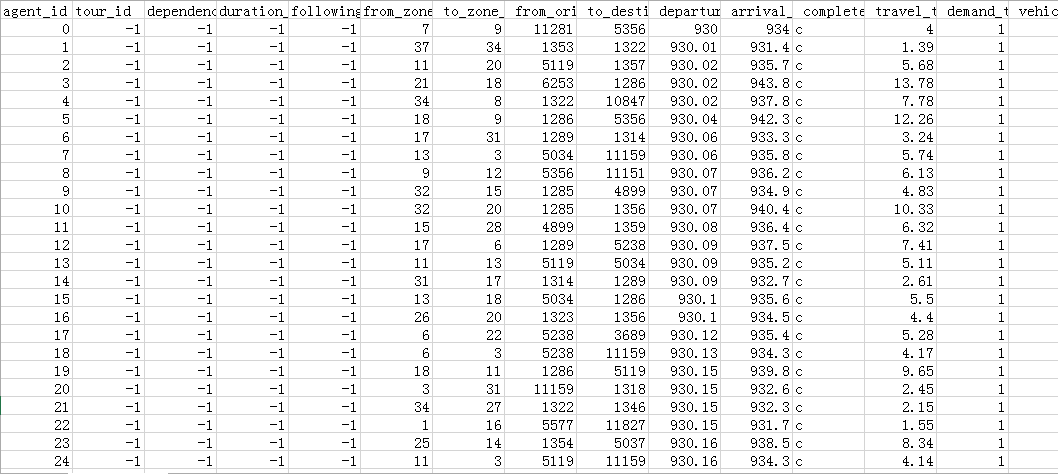
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **file\_name** | **format\_type** | **number\_of\_lines\_to\_be\_skipped** | **loading\_multiplier** | **start\_time\_in\_min** | **end\_time\_in\_min** | **demand\_type\_1** |
| demand.dat | dynasmart | 0 | 1 | 360 | 660 | 1 |
| demand\_hov.dat | dynasmart | 0 | 1 | 360 | 660 | 2 |
| demand\_truck.dat | dynasmart | 0 | 1 | 360 | 660 | 3 |

Each record corresponds a file for a certain demand type, using the same format\_type = dynasmart but different demand\_type\_1 = 1,2,3 for LOV, HOV and trucks. We do not need to skip the first line so number\_of\_lines\_to\_be\_skipped = 0, As DTALite uses a 24-hour time clock for both input and output format, we map the demand loading period to 6AM (start\_time\_in\_min = 360 min) and 11AM (end\_time\_in\_min = 660.).

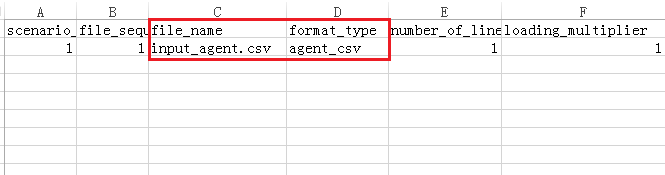
Please note that, the existing Dynus-T package uses the same demand format as the DYNASMART-P package, so if you have a Dynus-T data set, please use the above demand meta data settings directly.

#### Example 6: Agent file after OD demand estimation run

The OD estimation process will calibrate path flow pattern according to the observed link counts and density measurements, and the resulting calibrated results (at the last iteration of ODME) are saved as a binary file as agent.bin file, and an ASICII file called output\_agent.csv.



To evaluate traffic measurement strategies, after the ODME run, one can rename output\_agent.csv to input\_agent.csv, and set format\_type= agent\_csv in input\_demand\_file\_list.csv.



Run regular traffic assignment again (e.g. set number of iterations = 1, and using traffic assignment method = MSA or fixed switching rate).



#### input\_scenario\_settings.csv

The scenario settings file allows the user to alter the characteristics of the scenarios being run, as well as create various traffic scenarios that can be run simultaneously. Scenario attributes such as demand multiplier, traffic flow model, and number of days a scenario will be run can all be changed in this file. Further, each row can contain data for a separate scenario, allowing the user to simultaneously run models with differing model attributes. The scenario settings file allows the user to alter different attributes for each scenario. Starting from the far-right column, these attributes are:

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Example Usage** |
| scenario\_no | This is a discrete integer value assigned to a given scenario, and will be used as the scenario’s unique identifier when the simulation is running in DTALite. | scenario\_no =1 |
| scenario\_name | This is the identifier by which the scenario will be displayed to the end user. This identifier, unlike the scenario\_no, need not be an integer. | scenario\_name = test1 |
| number\_of\_assignment\_days | This value, an integer, is the number of days the scenario will be run. If the user is employing Origin-Destination Matrix Estimation, the scenario should run for at least 15 assignment days. | number\_of\_assignment\_days = 95 |
| demand\_multiplier | This value is the number by which the demand given in the input\_demand.csv file will be multiplied for a given scenario, e.g. if the demand for a given OD pair is 1000, and a demand multiplier of 1.8 is used for a given scenario, then for that scenario DTALite will use a value of 1800 for the demand on that OD pair. | demand\_multiplier =0.7 |
| random\_seed | This value is the seed number used for the pseudorandom number generator, used to create a level of randomness in certain aspects of the simulation. | random\_seed =100 |
| traffic\_flow\_model | This value must be one of four possible values:  0: BPR function  1: Point Queue Model. This model assumes vehicles “stack up” at nodes, rather than filling up the link.  2: Newell’s N-Curve Model. The most thorough of the four models, which takes into account features such as wave propagation through traffic.  3: combine BPR function and point queue model. Traffic flow model of each link can be further defined as BPR function or point queue model in **inpu\_link.csv** | traffic\_flow\_model =1 |
| freeway\_bias\_factor | This value dictates the degree to which agents modeled in the simulation will choose routes. The default value is sufficient for most simulations. | freeway\_bias\_factor=1 |

#### optional\_MOE\_settings.csv

The measure of effectiveness, or MOE, settings allow the user to test the effectiveness of the network as a whole, or smaller sections of a network, such as a single link, 3-point path, or origin-destination pair. The MOE settings file also allows the user to identify links, paths, and origin-destination pairs that are above user-defined threshold values. The following are the possible values for the MOE\_type field (the first column in the optional\_MOE\_settigns.csv file), as well as which fields must be filled in for each:

|  |  |
| --- | --- |
| **MOE\_type** | **Description** |
| network | Network MOE measures the effectiveness of the network at large. This network-wide measure can also be broken down based on attributes such as demand type and vehicle type. |
| od | This MOE type gauges the effectiveness of the network from one zone to another. The only field that must be populated are “origin\_zone\_id” and destination\_zone\_id It should be noted that it will only measure the effectiveness in the from-to direction. That is to say, if zone 1 is set as the origin zone, and zone 2 as the destination, effectiveness will only be measure from zone 1 to 2, not 2 to 1. In order to measure effectiveness in both directions, create two separate OD MOEs. |
| link | To measure the effectiveness of a link, only the “from\_node\_id” and “to\_node\_id” fields must be populated. As with OD MOE, the measure of effectiveness only goes in the from-to direction. |
| path\_3point | Much the same as link MOE, 3-point path MOE measures the effectiveness of a path between three connected nodes. This MOE needs an entry in “from\_node\_id,” “mid\_node\_id,” and “to\_node\_id.” |
| network\_time\_dependent | This measures the effectiveness of the network on a minute-by-minute basis. The results from this MOE are displayed in the output\_NetworkTDMOE.csv file. |
| od\_critical, link\_critical, path\_critical | Each of these only requires the user to enter a value in the “threshold\_volume” field. For example, if link critical MOE is performed with a threshold volume of 1250, then, in the output summary file, NeXTA will print MOE results for all of the links with volume over 1250. |

**Remark:**

1. The “moe\_group” column is used to break the MOE settings into discreet groups in the output summary. For example, to have all MOE critical values displayed together, assign them all the same group number, and they will be clustered together in the output summary.
2. The “moe\_cetegory\_label” is a user-defined field used to give simpler names to each individual measure of effectiveness. Each MOE may also have an associated start and end time based on when vehicles enter or exit the network, OD pair, link, or path.

### Scenario Files

The user may prepare scenarios by preparing the following input files, which describe different network conditions so that their effects on operations may be evaluated. Different scenarios available include tolling (distance-based and link-based tolls), dynamic message signs, incidents, and work zones.

#### Scenario\_Work\_Zone.csv

The work zone scenario input file is used to define the location and characteristics of work zones in the simulation, which is described in terms of capacity reduction, project duration, and speed reduction. This file can be used to model the impact of incidents, ramp meterings and traffic signal control devices.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Type** | **Acceptable Values** | **Description** |
| Link | Integer | [1,2] | Node pair [upstream, downstream] used to identify the link on which work zone is located |
| Day No | Integer | Value > 0 | Day identification number in the simulation on which the work zone causes capacity reductions |
| Start Time in Min | Integer | 0 to 1440 | Starting time for capacity reduction due to work zone |
| End Time in min | Integer | 0 to 1440 | Ending time for capacity reduction due to work zone |
| Capacity Reduction Percentage (%) | Float | Value ≥ 0 | Capacity reduction percentage (1 – percent remaining capacity) due to work zone |
| Speed Limit (mph) | Integer | Value ≥ 0 | Speed limit on link posted during work zone |

#### Scenario\_Dynamic\_Message\_Sign.csv

The dynamic message sign scenario input file is used to define the location and characteristics of variable message signs in the simulation, which influences driver route choice by the response percentage defined in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Type** | **Acceptable Values** | **Description** |
| Link | Integer | [1,2] | Node pair [upstream, downstream] used to identify the link on which the sign is installed |
| Start Time in Min | Integer | 0 to 1440 | Starting time for the dynamic message sign display |
| End Time in min | Integer | 0 to 1440 | Ending time for the dynamic message sign display |
| Number of Detour Routes | Integer | Value ≥ 0 | Percentage of drivers on the link which respond to the real time information displayed on the sign. |
| Detour Route 1 | String | 2;2;3;0.60 | The first “2” means the route 1 is defined by 2 nodes, node 2 and node 3, represented by “2;3”. “0.60” means that 60% of travelers at node 2 will choose route 1. The starting node of two routes is node 2, which is the downstream node of link [1, 2] where the vms is located.  For route 2, the same explanation is also applied. |
| Detour Route 2 | String | 2;2;4;0.40 |

#### Scenario\_Link\_Based\_Toll.csv

The link-based toll scenario input table is used to define tolling conditions on a road segment in the simulation. Currently, there are three classes defined for different toll pricing – SOV, HOV, and trucks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Type** | **Acceptable Values** | **Description** |
| Link | Integer | [1,2] | Node pair [upstream, downstream] used to identify the link on which the toll is implemented |
| Day No | Integer | Value > 0 | Day identification number in the simulation on which the tolling strategy is implemented |
| Start Time in Min | Integer | 0 to 1440 | Daily starting time for the link-based toll |
| End Time in min | Integer | 0 to 1440 | Daily ending time for the link-based toll |
| Toll for Demand Type 1 ($) | Float | 0 to 999 | Charge for first demand type to travel across the link |
| Toll for Demand Type 2 ($) | Float | 0 to 999 |  |
| Toll for Demand Type N ($) | Float | 0 to 999 | Charge for the last demand type to travel across the link, the user should make sure the number of demand types is consitsent with the total number of demand types defined in input\_demand\_type.csv |

### Output Files

The following table summarizes the output files from DTALite traffic assignment.

|  |  |  |
| --- | --- | --- |
| **File Name** | **Type** | **Data Description** |
| [Output\_summary.csv](https://docs.google.com/spreadsheets/d/14P82EeRpBLBZT3VVKS1lY3MpcCFsCLu8ZW8dQAhGL4M/edit" \l "gid=2119997585) | Scenario statistics | Traffic assignment results for each iterations, such as computational time, average travel distance and travel time, number of travelers and user equilibrium gaps. |
| [Output\_NetworkTDMOE.csv](https://docs.google.com/spreadsheets/d/1shISC0ASaA784fFkqAmvWrEpE8G-56D30NTzKQdKDYE/edit" \l "gid=1865952138) | Network | Time-dependent network-level information about assignment results over the modeling horizon, such as, cumulative inflow count, cumulative outflow count, number of vehicles existing in the network, average trip time. |
| [Output\_ODMOE.csv](https://docs.google.com/spreadsheets/d/12WRLPaVm5BfHp7xC3sbxoQTrttNNe1Pgu4foNmGeVB4/edit" \l "gid=1228094423) | OD | ODMOE simulation results, such as, number of vehicles completing trips, trip time, trip distance for each OD pair. |
| [Output\_linkMOE.csv](https://docs.google.com/spreadsheets/d/1S18Pf-D8F5fFAz7PaCK93nbhliTHw3SUcbX8a41L9qI/edit" \l "gid=1062729698) | Overall link statistics for all links in the entire network | Detailed simulation results aggregated at each link, such as, link volume, link speed, level of service, emissions, volume of different types of vehicles. |
| [Output\_linkTDMOE.csv](https://docs.google.com/spreadsheets/d/1gToOECJqT2T2rDpS6YSPRXZvz-xi4iN2iV32nrjK-AQ/edit" \l "gid=2142282830) | Time-dependent link statistics for for all links in the entire network | Time-dependent detailed simulation results aggregated at each link at each minute, such as, density, volume, speed, queue length, emissions.  Output\_linkTDMOE.bin is read for the time-dependent network-level visualization. |
| [Output\_agent.csv](https://docs.google.com/spreadsheets/d/1Jrmq3tlV22qrwJb4tzHivMio04DNHZwHqdlJOcE_oIc/edit" \l "gid=689476001) | Vehicle/ agent | Specific information of each agent in the simulation network, such as, origin, destination, departure time, node sequence of its path, time sequence of each node in its path, emissions;  NEXTA uses binary file for fast data processing |

# USER INTERFACES & BASIC CONTROLS

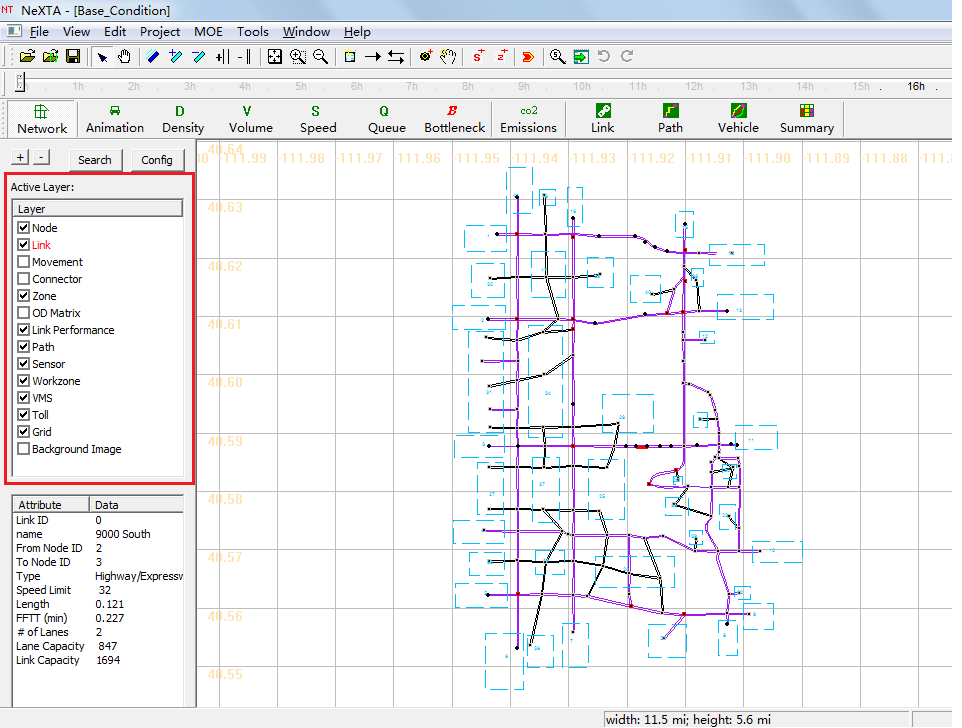
## User Interfaces

### GIS Interface

The latest version of NeXTA uses a GIS-like user interface to add functionality/flexibility while providing an interface which is similar to software applications commonly used in transportation modeling. In general, the GIS interface in NeXTA is described by its layer and visualization controls, allowing the user to modify which object types are shown on the screen and how those objects are displayed.

### Layer Control Panel

NeXTA’s user interface uses layer controls which are similar to those used in common GIS software applications to manage which network object types are displayed/selected.



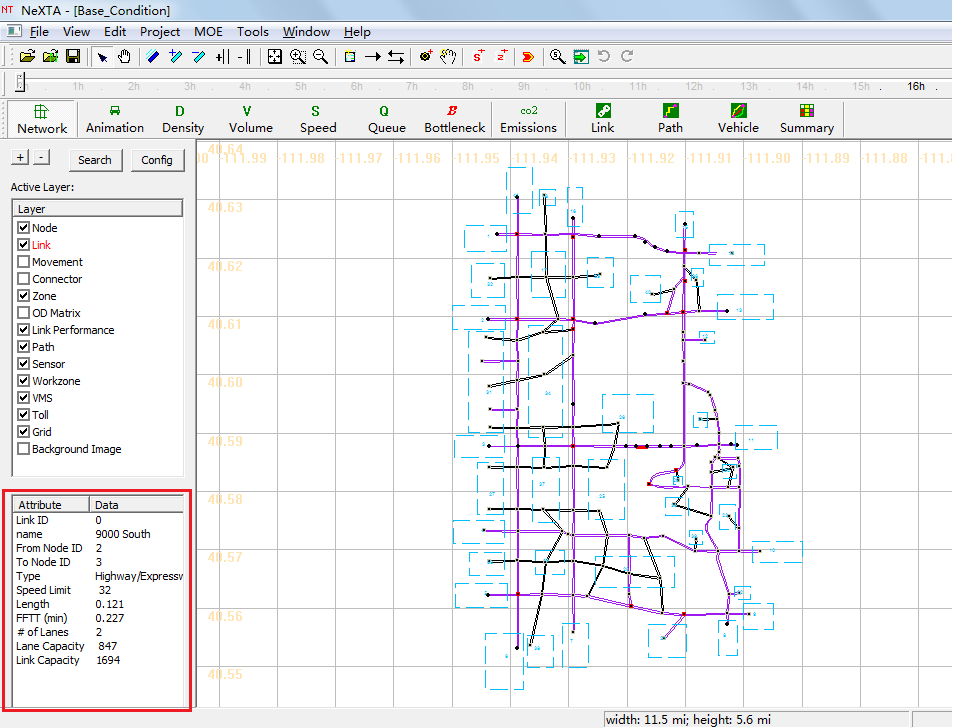
The list of layers at the left side of the screen, highlighted in the figure above, is used to control what is visible in the display. The panel display controls the Node, Link, Movement, Connector, Zone, OD Matrix, Link Performance, Path, Sensor, WorkZone, VMS, Toll, Grid and Background Image layers. Each layer refers to a different type of network data, which is stored in the network input/output files in the project folder.

**The box alongside each layer’s text label** is used to control the layer’s visibility. An empty box indicates that the layer is not visible, and a check mark in the box indicates that the layer is visible (if data is available for display). In some cases, after turning a layer on or off, the user may need to click the layer’s text label to refresh the display for that specific layer.

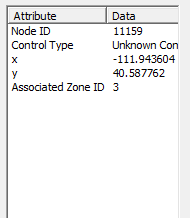
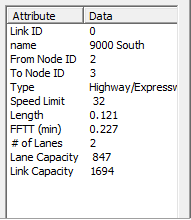
**The layer text label** is used to control which objects can be selected in the network. In particular, the selectable layers are limited to the layers such as Node, Link, Movement, Connector, Zone and OD Matrix. With the layer turned on (enabled), left-clicking on the layer text label enables selection using the Select Object tool. The text label is highlighted in red text after selection, indicating which network object type can be selected using the Select Object tool. Please see more detail illustration for examples for using this functionality.

### Attribute Data Display Panel

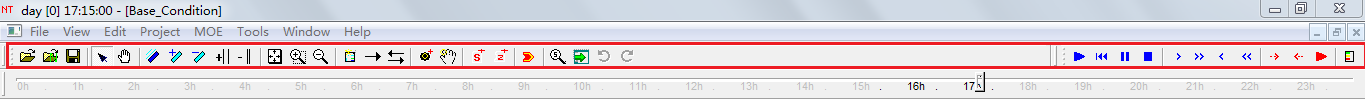
The lower half of the panel at the left side of the screen shows attribute data for a selected object, as shown in the figure below. The information displayed in this section of the panel is dependent upon the selected network object type.



Node attribute data displayed in the panel includes the node ID number, control type, geographic coordinates, and associated zone ID number (=corresponding zone number, if a node is an activity location; =0, otherwise) for the selected node. Link attribute data displayed in the panel includes the link ID number, link name, starting node ID number, ending node ID number, link type, speed limit, length, free-flow travel time, number of lanes, lane capacity and link capacity for the selected link. An example is shown in the figure below with link attribute data.

## Management Toolbar



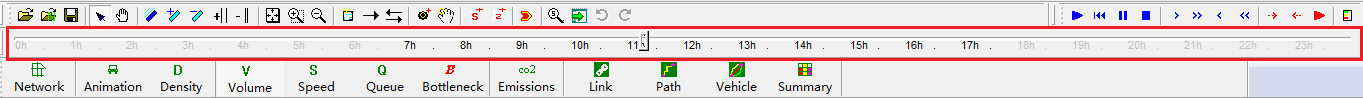
### Basic Management Tools

|  |  |  |
| --- | --- | --- |
| **Icon** | **Name** | **Function** |
|  | Open Project |  |
|  | Open New Project |  |
|  | Save Project | Save network to given path/file name |
|  | Select Object | Select a node/link/zone |
|  | Move Network |  |
|  | Switch Link Bar/Line Display Mode |  |
|  | Increase Link Bandwidth |  |
|  | Decrease Link Bandwidth |  |
|  | Increase Link Offset |  |
|  | Decrease Link Offset |  |
|  | Show Network | Show entire network |
|  | Zoom In |  |
|  | Zoom Out |  |
|  | Search Node/ Link/Path/Vehicle | Opens a dialog box which enables search functionality in NeXTA. Search by node number to find nodes, links (from node and to node notation), paths (from node and to node notation, using shortest path), and vehicle number (when simulations results are available). |
|  | Visit Development Website |  |
|  | Run Simulation |  |

### Network Editing Tools

|  |  |  |
| --- | --- | --- |
| **Icon** | **Name** | **Function** |
|  | Set Default Link Type | Opens a dialog box displaying the default link properties for different link types. The user may select and edit the default link properties so that all new links created afterward are assigned those changes. |
|  | Add New One-Way Links | Create a new one-way, directional link between two nodes. |
|  | Add New Two-Way Links | Create two one-way, directional links between two nodes. |
|  | Add New Node | Create a new node to which links can be attached. |
|  | Move Node Position |  |
|  | Create Subarea for subarea Analysis | Create a subarea boundary which is used to perform a subarea cut (see Subarea Analysis for more details). |
|  | Create New Zone | Create a new zone |

### Time Controlling Tools



**The Clock Bar**(highlighted in the figure above) is a toolbar feature located at the top of the screen which allows the user to view time-dependent MOEs by controlling the position of the slider on the toolbar. As shown above, the toolbar is divided into hours so that the position of the slider refers to the time within a 24-hour modeling time horizon.

The buttons at the right above the Clock Bar are used for controlling the progression of time. This can also be accomplished by using the mouse to move the slider, clicking and dragging the slider to the desired location (time) on the bar.

|  |  |  |
| --- | --- | --- |
| **Icon** | **Name** | **Function** |
|  | Star Animation (Min by Min) | Progresses forward automatically through time in 1 minute steps |
|  | Rewind | Rewind the time back to 00:00 |
|  | Pause Animation | Temporarily stops the automatic progression of time until the play button is pressed again |
|  | Stop Animation | Stops the automatic progression of time and reset the time back to 00:00 |
|  | Play Forward 1 Min | Moves forward in time by 1 minute |
|  | Skin Forward 5 Min | Moves forward in time by 5 minutes from the current time |
|  | Play Backward 1 Min | Moves backward in time by 1 minute |
|  | Skin Backward 5 Min | Moves backward in time by 5 minutes from the current time |
|  | Play Forward 1 Sec | Moves forward in time by 1 second |
|  | Play Backward 1 Sec | Moves backward in time by 1second |
|  | Star Animation (Sec by Sec) | Progresses forward automatically through time in 1 second steps |
|  | Show/Hide Legend | Toggles legend visibility |

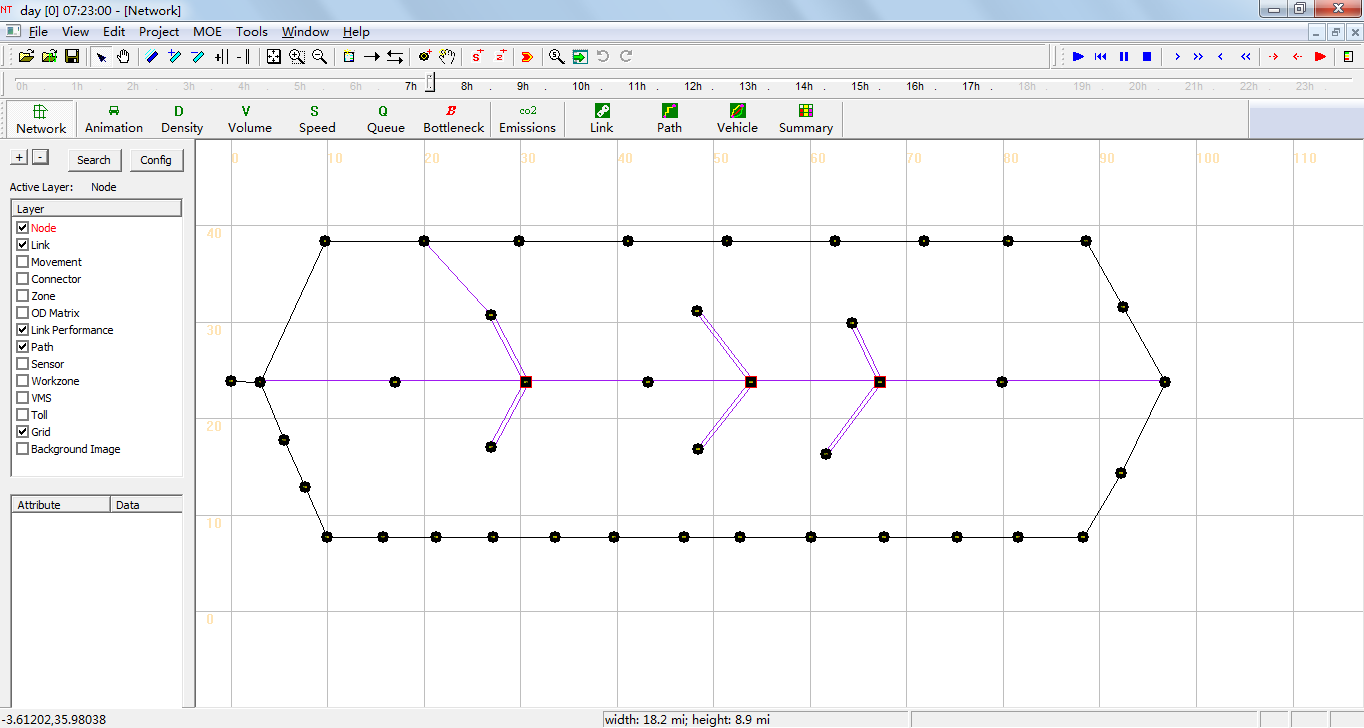
## Viewing Modes



Two different viewing modes are available in NeXTA – Network View mode and Animation View mode. The default Network View mode is used to display Measures of Effectiveness (MOEs) and the network geometry, while the Animation View is used to show individual vehicles moving in the network during simulation. The user can use the and  buttons on the MOE Toolbar to control which view is used.

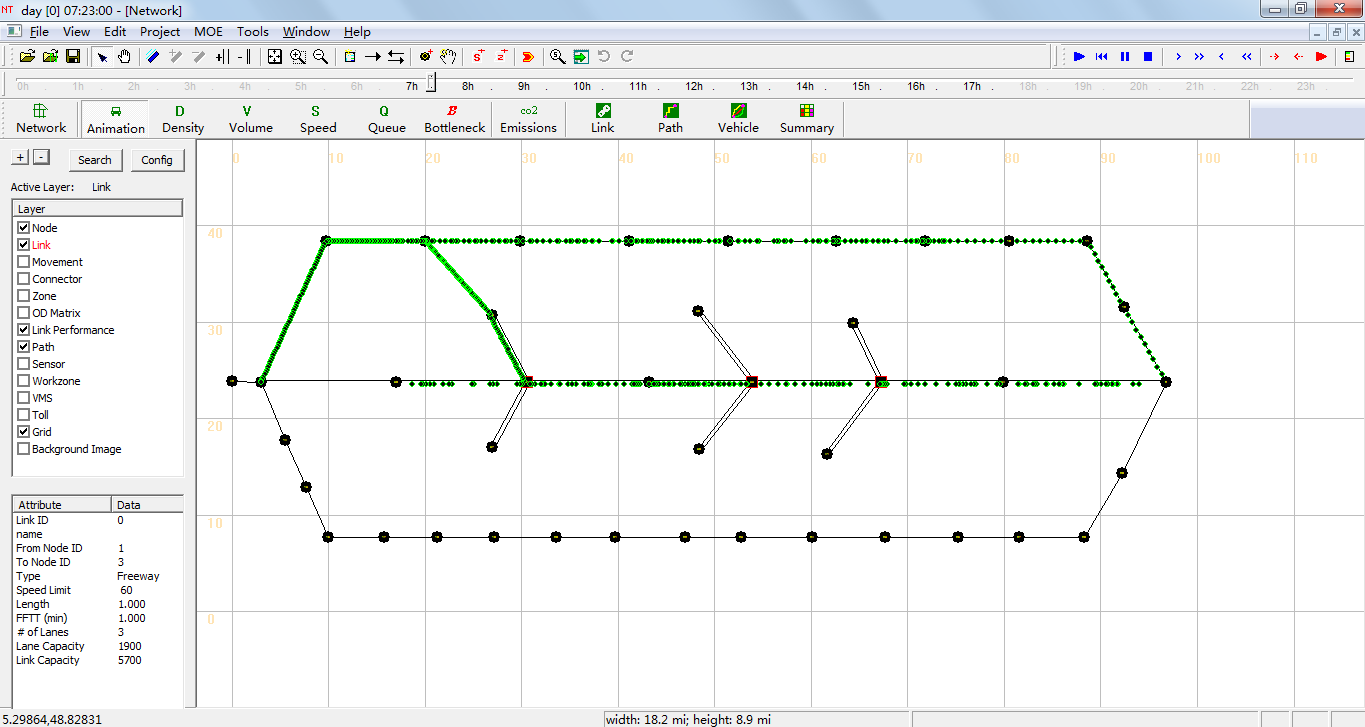
### Network View

In the default visualization state, each link is shown with a line width to represent the number of lanes. Additionally, many MOE visualization features use the link width to visually show how MOEs change over time or differ from one link to another. The  button on the MOE Toolbar changes this visualization state so that no links in the network will be shown with a link width.



### Animation View

The Animation View changes the visualization state to show vehicles moving in the simulation over time, where the time step is controlled by the Clock Bar. This feature is engaged by pressing the  button. Vehicles in the simulation are represented as green circles moving along the links in the network. An example is provided below, showing a portion of the 3-Corridor network sample network.



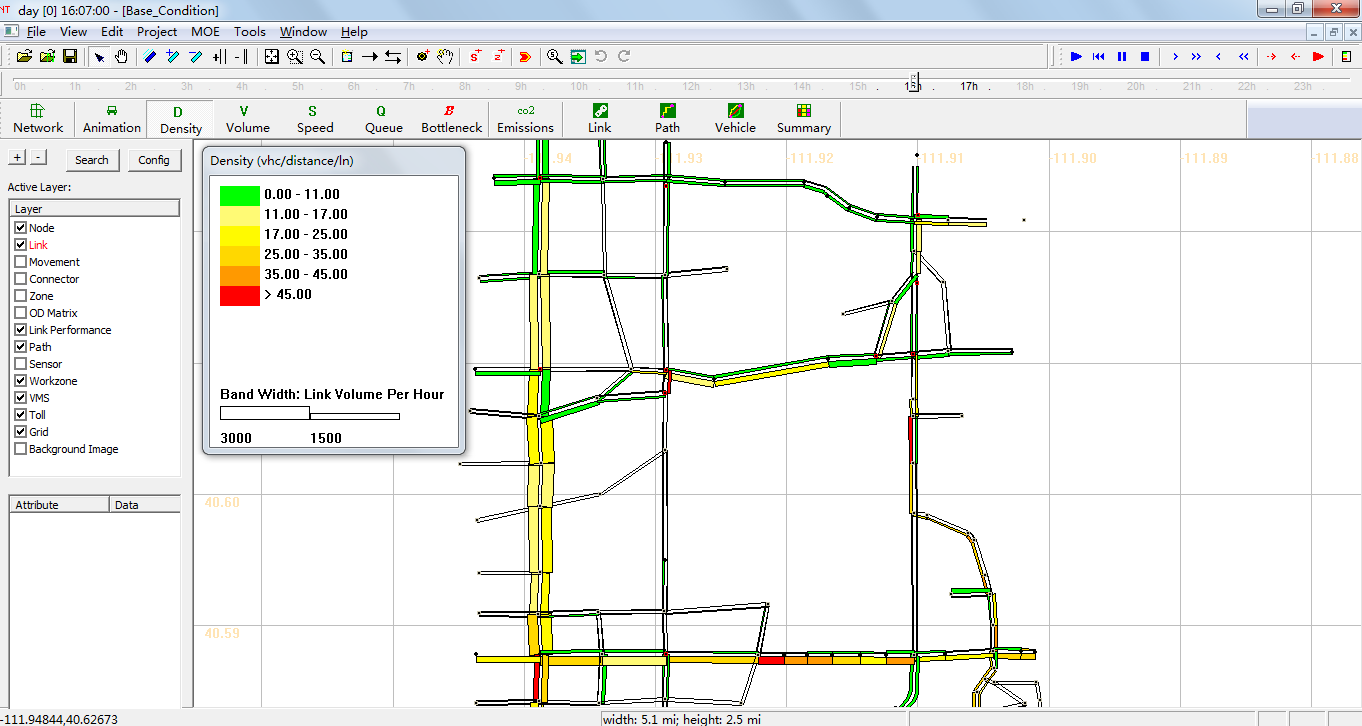
## MOE Toolbar



The majority of the visualization tools provided in NeXTA are available through the MOE Toolbar features highlighted below. Traditional MOE (Measure of Effectiveness) visualizations are provided for the Density, Volume, Speed and Queue. In general, multiple visualization modes cannot be enabled at the same time, except in the case of using the Synchronized Display Mode. In addition to these traditional visualization options, three new analysis features are currently available: Bottleneck and Emissions. Each visualization feature is explained in the relevant sections below.

### Density Visualization

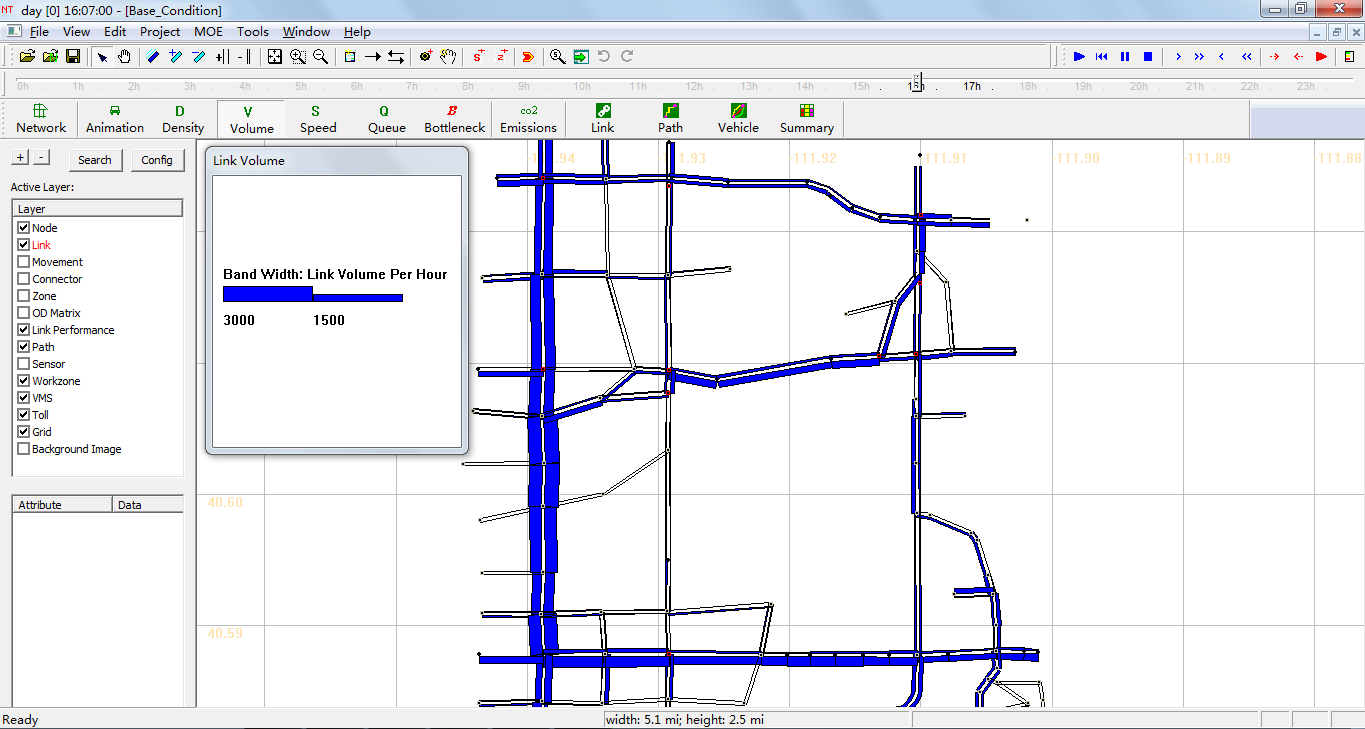
The Density Visualization View is enabled using the  button, showing the time-dependent density for each link in the network. An example is shown below for a portion of the West Jordon network, where the link width is based on the time-dependent link volume. The visualization may be modified to show links without widths by using the  button.



Each link is color-coded based on the MOE value at the time specified by using the Clock Bar, and a legend is provided (shown at left in the above figure) to relate MOE values to color codes. Legend visibility can be toggled using the  button on the toolbar.

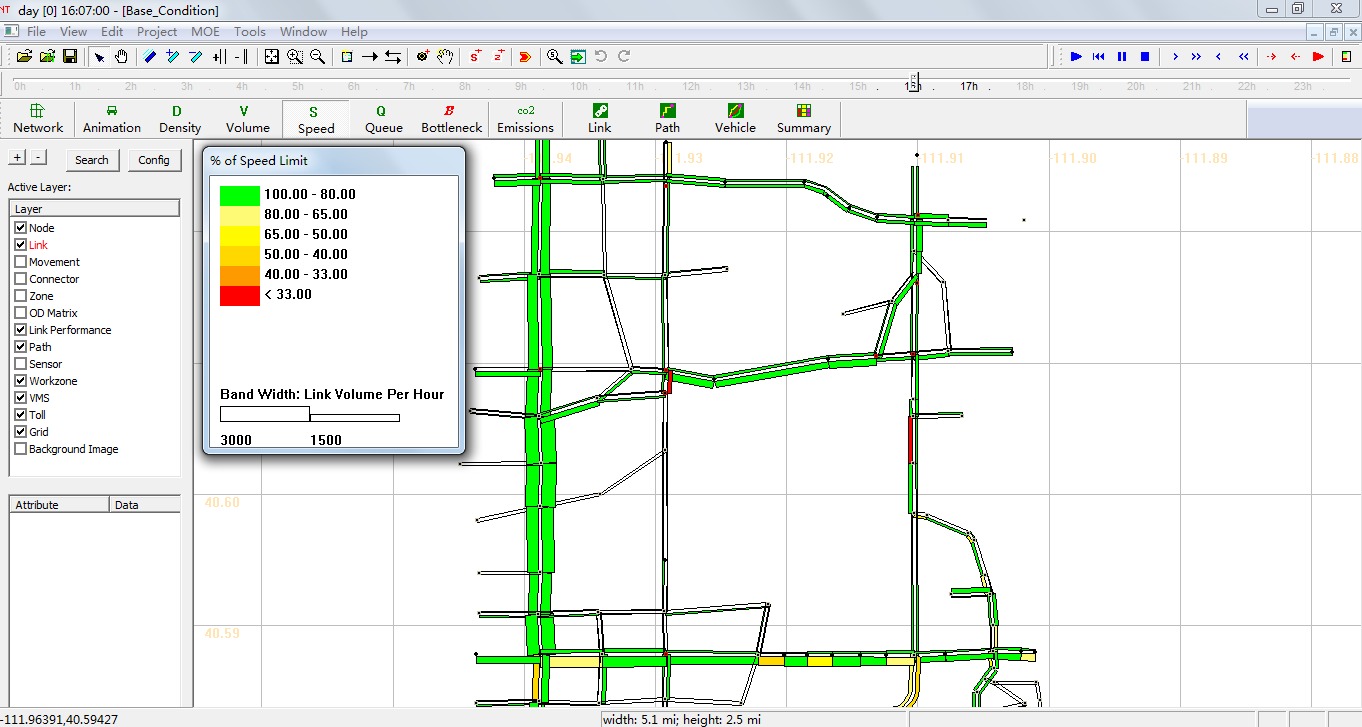
### Volume Visualization

The Volume Visualization View is enabled using the  button, showing the time-dependent volume for each link in the network. An example is shown below for a portion of the West Jordon network, where the link width is based on the time-dependent link volume. The visualization may be modified to show links without widths by using the  button.



### Speed Visualization

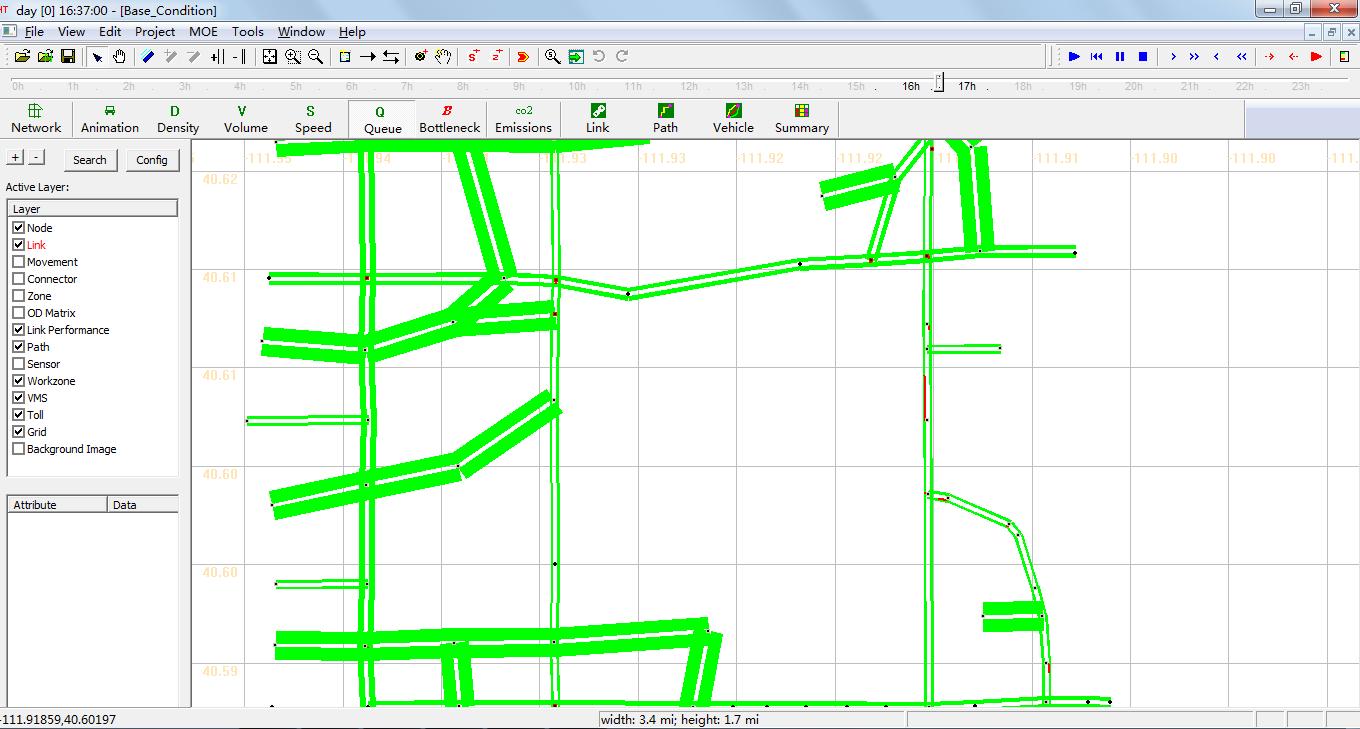
The Speed Visualization View is enabled using the  button, showing the time-dependent speed for each link in the network. An example is shown below for a portion of the West Jordon network, where the link width is based on the time-dependent link volume. The visualization may be modified to show links without widths by using the  button.



Each link is color-coded based on the MOE value at the time specified by using the Clock Bar, and a legend is provided (shown at left in the above figure) to relate MOE values to color codes. As shown in the legend, the color coding is based on the ratio of the average speed vs. the specified speed limit for each specific link. Legend visibility can be toggled using the button on the toolbar.

### Queue Visualization

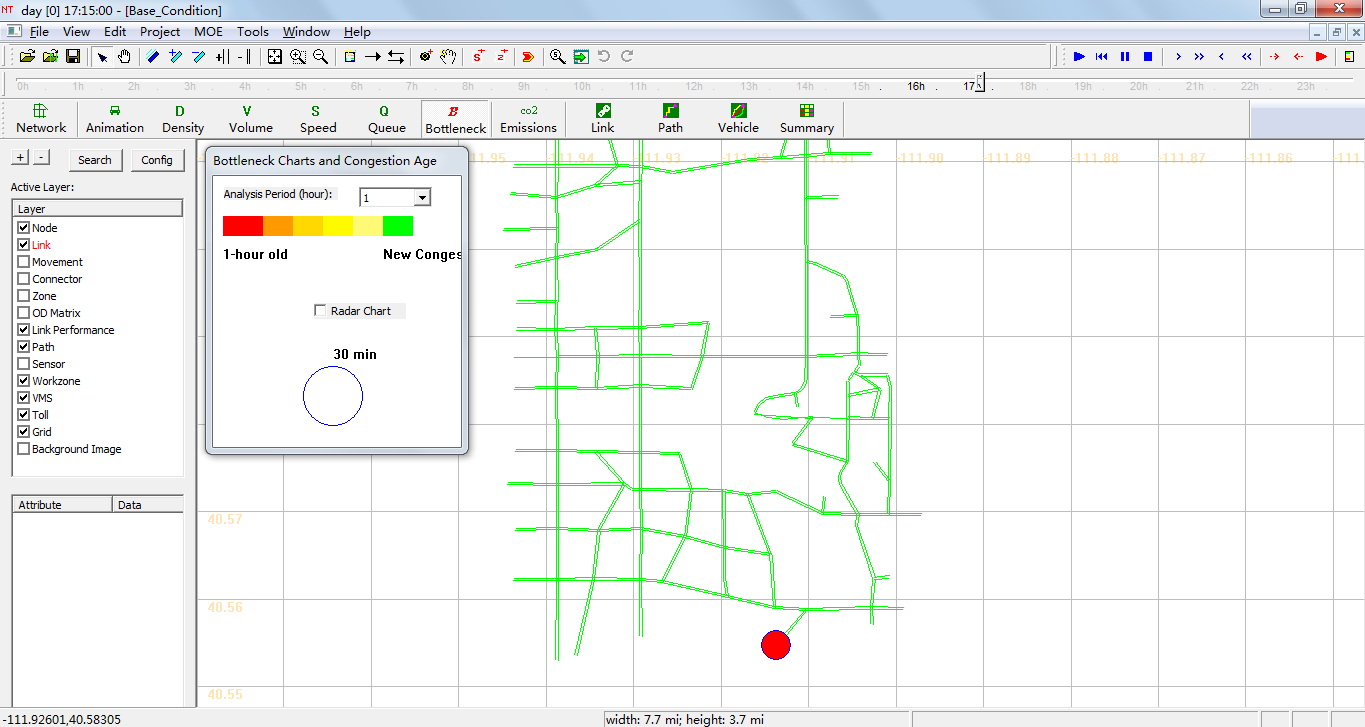
The Queue Visualization View is enabled using the  button, showing the time-dependent queue length for each link in the network. An example is shown in the figure below. This visualization mode works in both Network View Mode and with link widths corresponding to the number of lanes (not volume), and these viewing modes can be toggled using the  button.



The queue is visually represented as the portion of the link which is colored red, and the distance over which the red color is applied on each link represents the percentage of the link which is occupied with queued vehicles. The length of the red portion on the link changes dynamically over time, corresponding to the time-dependent queue length.

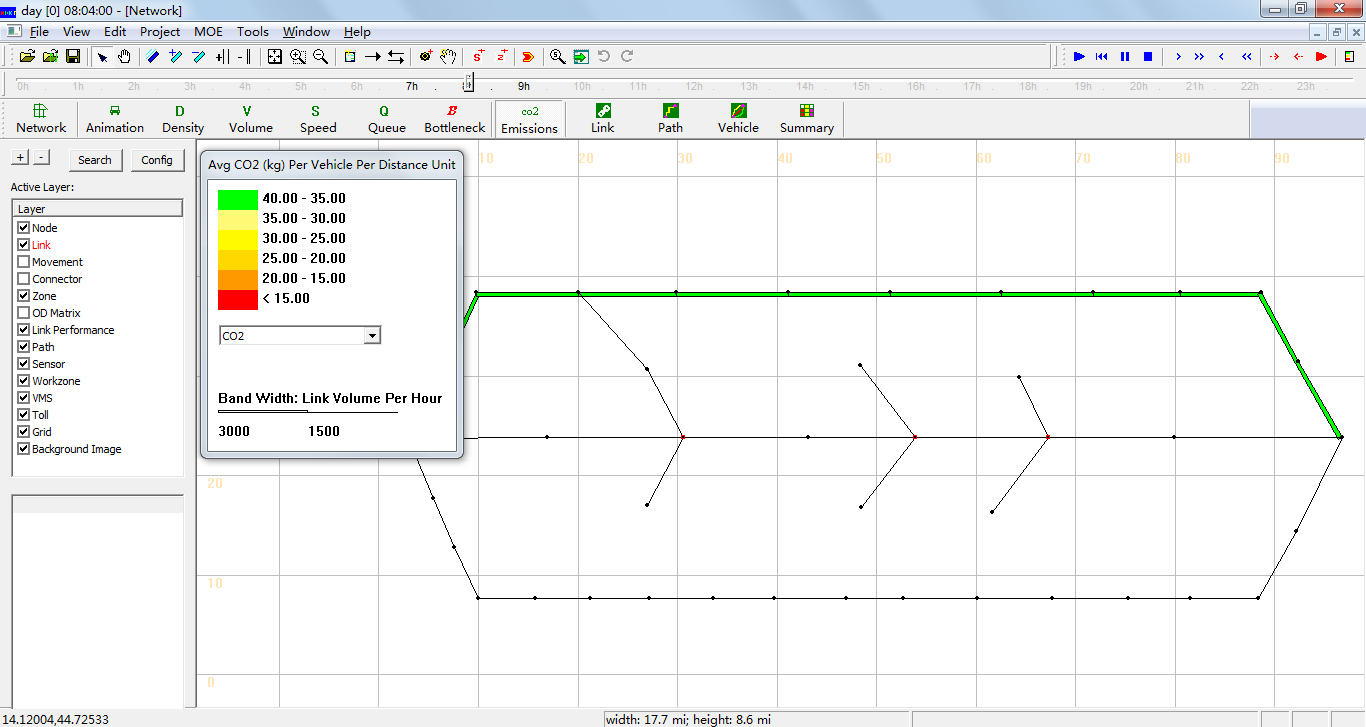
### Bottleneck Visualization

The Bottleneck Visualization View is enabled using the  button, showing the time-dependent congestion nodes in the network. An example is shown below for a portion of the West Jordon network. This visualization mode works in both Network View Mode and with node sizes corresponding to the age of congestion in analysis period, and these viewing modes can be toggled using the  button.

The bottleneck is visually represented as the portion of the congestion node which is color-coded, and the size of each congestion node changes dynamically over time, corresponding to the age of congestion. Legend visibility can be toggled using the button on the toolbar.

### Emissions Visualization

The Emissions Visualization View is enabled using the  button, showing the time-dependent emissions for each link in the network (only available in emission scenarios, more details in 4.1.3 Emissions Scenario sections). An example is shown below for a portion of the 3-Corridor network, where the link width is based on the time-dependent link volume, and these viewing modes can be toggled using the  button.



Each link is color-coded based on the emissions MOE value at the time specified by using the Clock Bar, and a legend is provided (shown at left in the above figure) to relate MOE values to color codes. Legend visibility can be toggled using the  button on the toolbar.

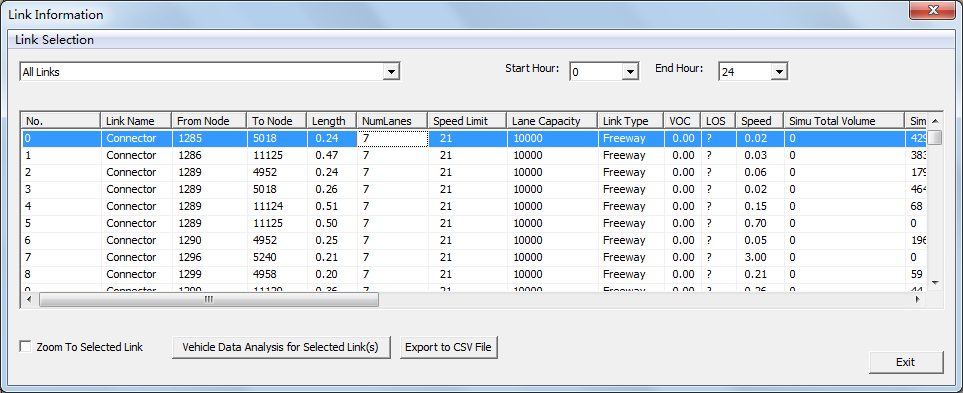
## Detailed Analytical Tools



In addition to the previously-described visualization tools, the Link, Path, Vehicle, and Summary analytical tools are available for more detailed analyses. These features may be accessed through the highlighted buttons shown below on the MOE Toolbar. Each visualization feature is explained in different sections below.

### Link Analysis Tool

Selecting the  button, or going to MOE > Link List Dialog, opening the Link Information window (as shown below), which is used to view link attributes and MOEs. Selecting a row with the mouse also selects the link in the network, allowing the user to quickly find specific links. The Link Zoom toggle button at the bottom left side of the window centers the network view window at the selected row after a row is selected. Each column of data can be used to sort the list, allowing the user to more quickly find links with specific attributes or which meet certain criteria. The “Vehicle Data Analysis for Selected Link(s)” button at the bottom of the window offers vehicle analysis data for selected links(more detail in 3.5.3 “Vehicle Analysis Tool” section). The “Export to CSV File” button at the bottom of the window helps users export link information and MOEs to CSV file.



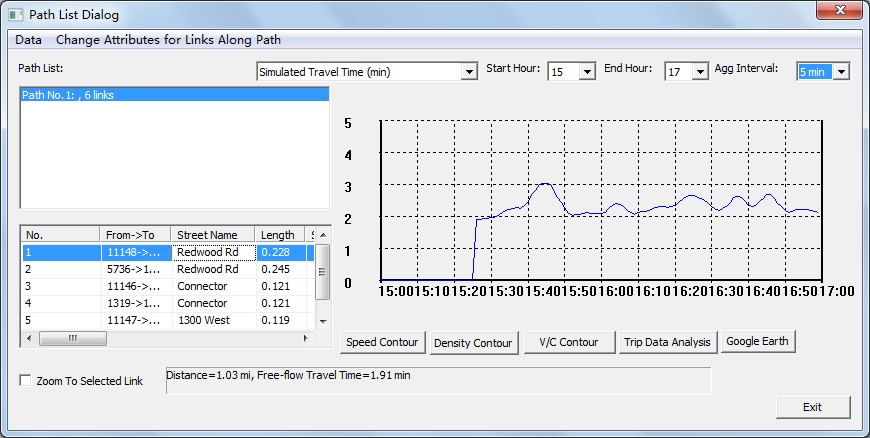
Additionally, the Link Selection menu at the top left side of the window offers options for filtering the rows by link type. Filtering options are available for displaying only Selected links, links within Subarea, Freeway, Highway, Ramp, Arterial, Connector links and Non-Connector links. The Start/End Hour Selection menu at the top right side of the window offer options for analysis period.

### Path Analysis Tool

The Path Analysis Tool is enabled by using the  button or going to MOE > Path List Dialog, which is used to view link attributes and path travel time statistics.

To use the tool, a path must first be selecting in the path layer. As a recap, this is accomplished by right-clicking the mouse at the origin node for the path, selecting “Direction from Here”, and then right-clicking again at the destination, selecting “Direction to Here”. The path is chosen automatically based on the shortest path between the two points. Additionally, An intermediate point can be chosen by selecting “add intermediate destination here”. You may also avoid intermediate nodes/links by selecting “avoid this node”/ “avoid using this link”.

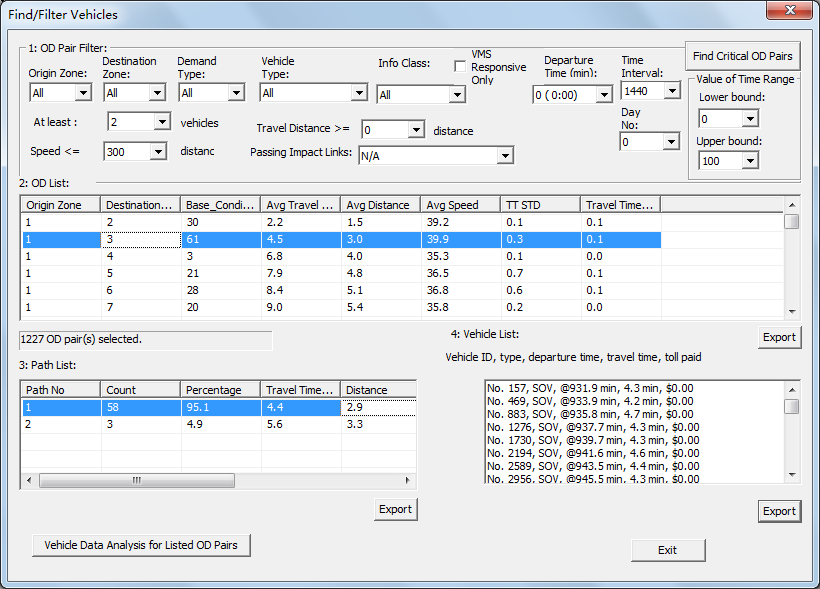
Selecting the  button opens the Path Information window, as shown in the example below. Similar to the Link Information window, this tool shows link attributes for the links in the path. Path statistics(including distance and free flow travel time) are shown in bottom of the window, and MOE values of path (including speed contour, density contour, V/C contour, Trip data Analysis and Google Earth) are shown in right side of the window.



Additionally, The Link Zoom toggle button at the bottom left side of the window centers the network view window at the selected row after a row is selected. Each column of data can be used to sort the list, allowing the user to more quickly find links with specific attributes or which meet certain criteria. The Start Hour, End Hour and Agg Interval Selection menu at the top right side of the window offer options for analysis period and time interval for statistics.

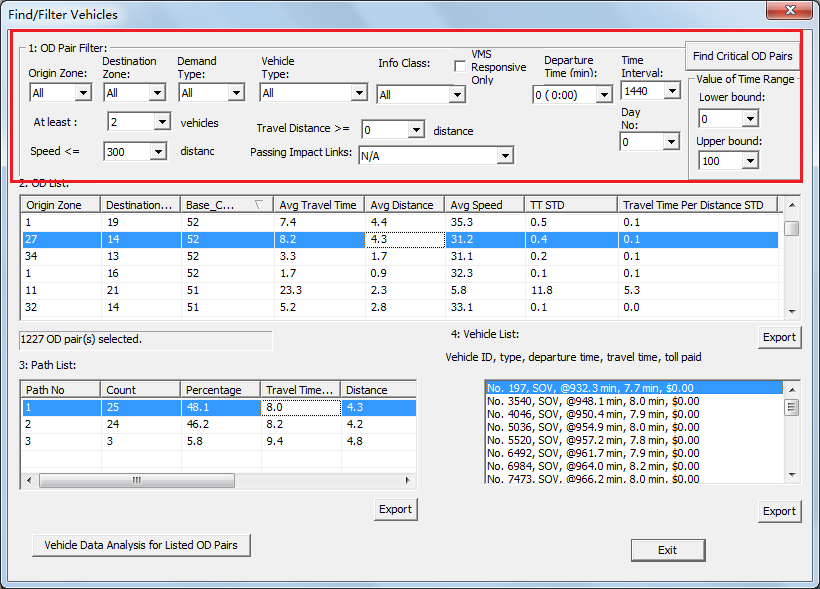
### Vehicle Analysis Tool

The Vehicle Analysis Tool, enabled using the  button or going to MOE > Vehicle Path Analysis, is a powerful analysis feature used to examine travel statistics for individual vehicle or groups of vehicles. This window is divided into four sections: the OD Pair Filter, OD List, Path List, and Vehicle List.

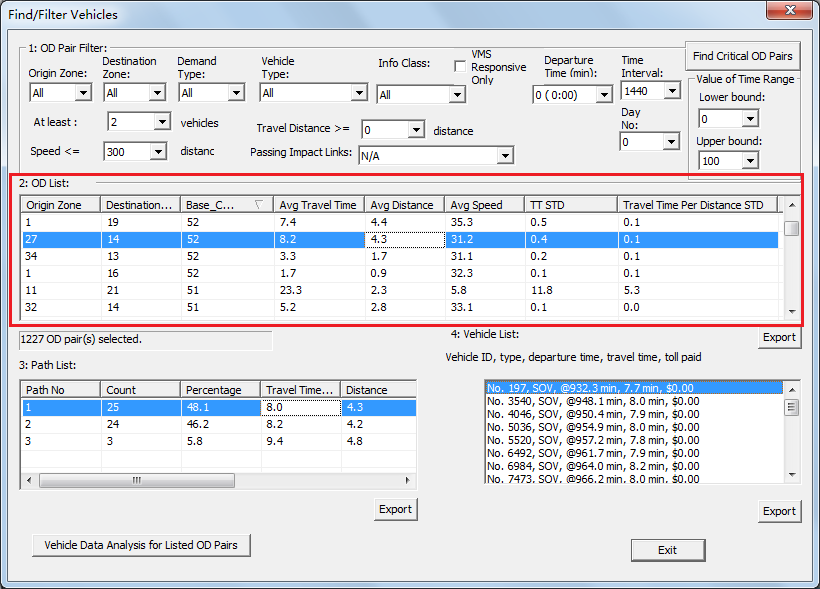


The OD Pair Filter (as shown in the figure below), located at the top of the window, offers several filtering options for limiting an analysis based on specific criteria. The top row of drop-down lists primarily provides filterable criteria related to the vehicle, including the Origin Zone ID, Destination Zone ID, Demand Type and Vehicle Type, Information Class and Departure Time(min), and whether the vehicle was traveling within a certain Time Interval. Also relevant to vehicle characteristics, a filter based on a range for the Value of Time is offered at the far right side of the window. The drop-down lists immediately to the right of the OD List are filterable criteria related to path attributes, including the Number of Vehicles using a path, the Total Travel Distance (in miles) and Travel Time Index on the path.

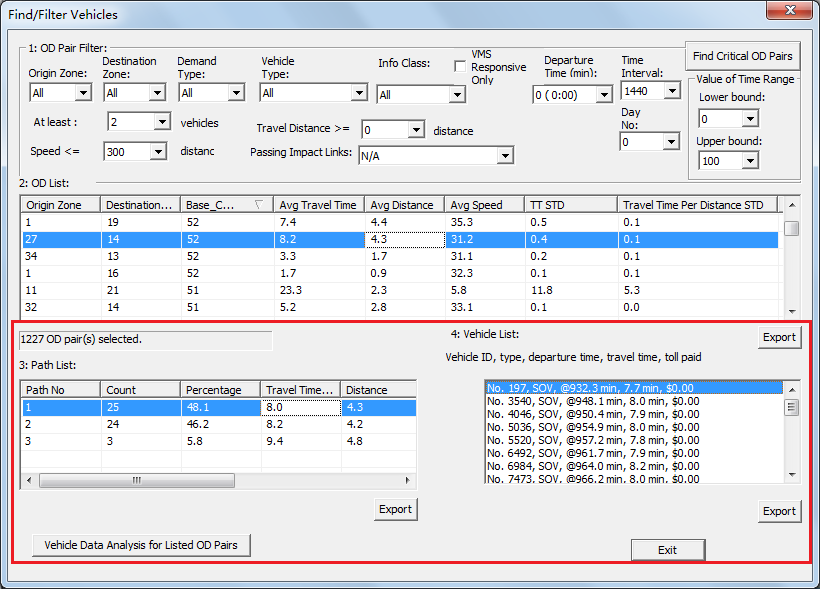
The Find Critical OD Pairs button, found at the top right corner of the window, uses some default filter criteria (path with more than 500 vehicles and at least 2 miles in length) to find the most important OD Pairs.



The OD List shows any Origin-Destination pairs which meet the criteria used in the OD Pair Filter (as shown in the figure below). Each pair is listed with origin and destination zone ID, along with the number of vehicles, average travel time, average distance, average speed, travel time STD and travel time per distance STD.

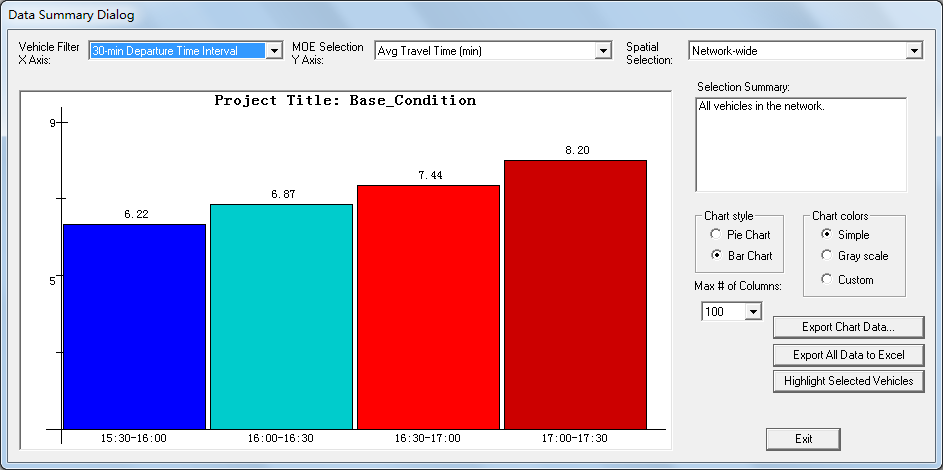


By clicking on a specific OD Pair in the OD List, the Path List and Vehicle List is populated with paths and vehicles associated with that specific OD Pair (as shown in the figure below). Selecting different paths in the Path List highlights those paths in the network, as shown below, and further limits the vehicles shown in the vehicle list to only those vehicles using the selected path. Export buttons are located near the bottom of each list so that the user may export the items in the separate lists and save them as CSV files.

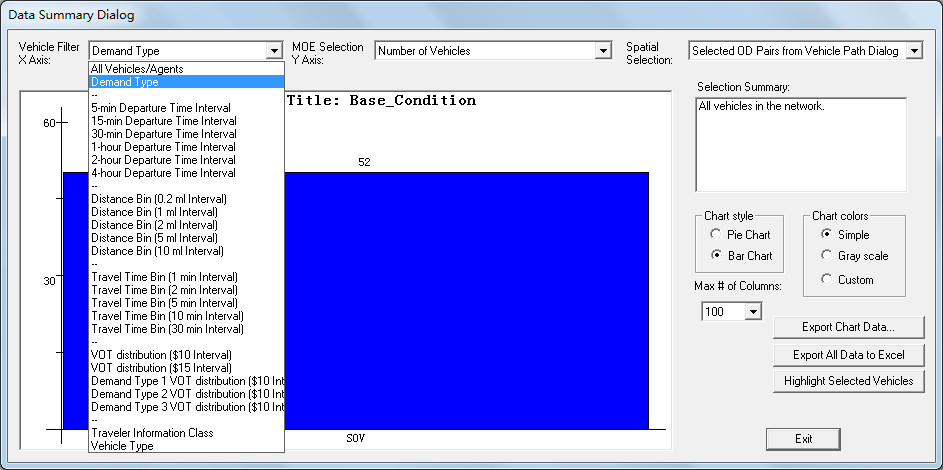


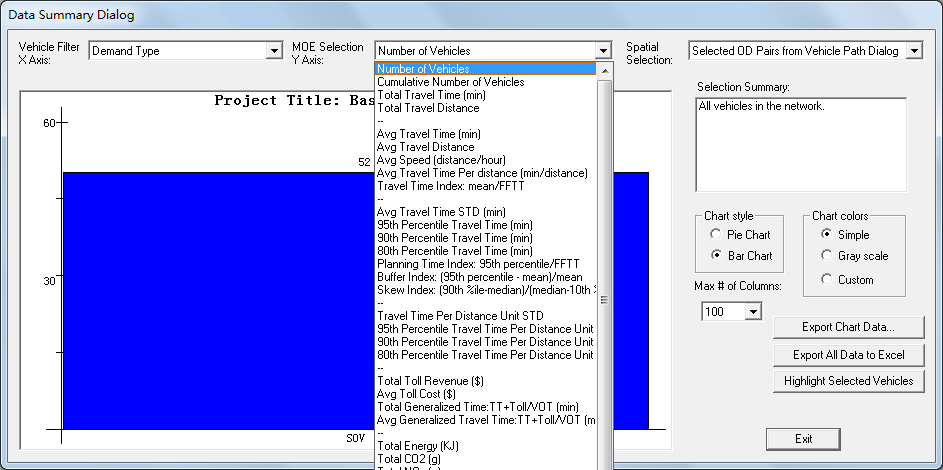
### Summary Analysis Tool

The System Analysis Tool is used to examine travel statistics for groups of vehicles. By using thebutton or going to MOE > Network Statistics Dialog, the Data Summary Dialog opens, as shown below.



The drop-down lists at the top of the window are inputs which allow the user to modify the chart shown in the window. As described in the figure, the first drop-down list controls the X Axis, and the second controls the Y Axis. The X Axis options (Vehicle Filter) will divide groups based on their Value of Time (VOT), vehicle type, demand type, traveler information class, departure time intervals, distance bin and travel time bin over the modeling horizon. The Y Axis options (MOE Selection) allow plotting the number of vehicles, total and average travel times, total and average travel distance, total toll revenue, average toll cost, total and average generalized travel cost (in minutes and dollars), total and average emissions and so on.

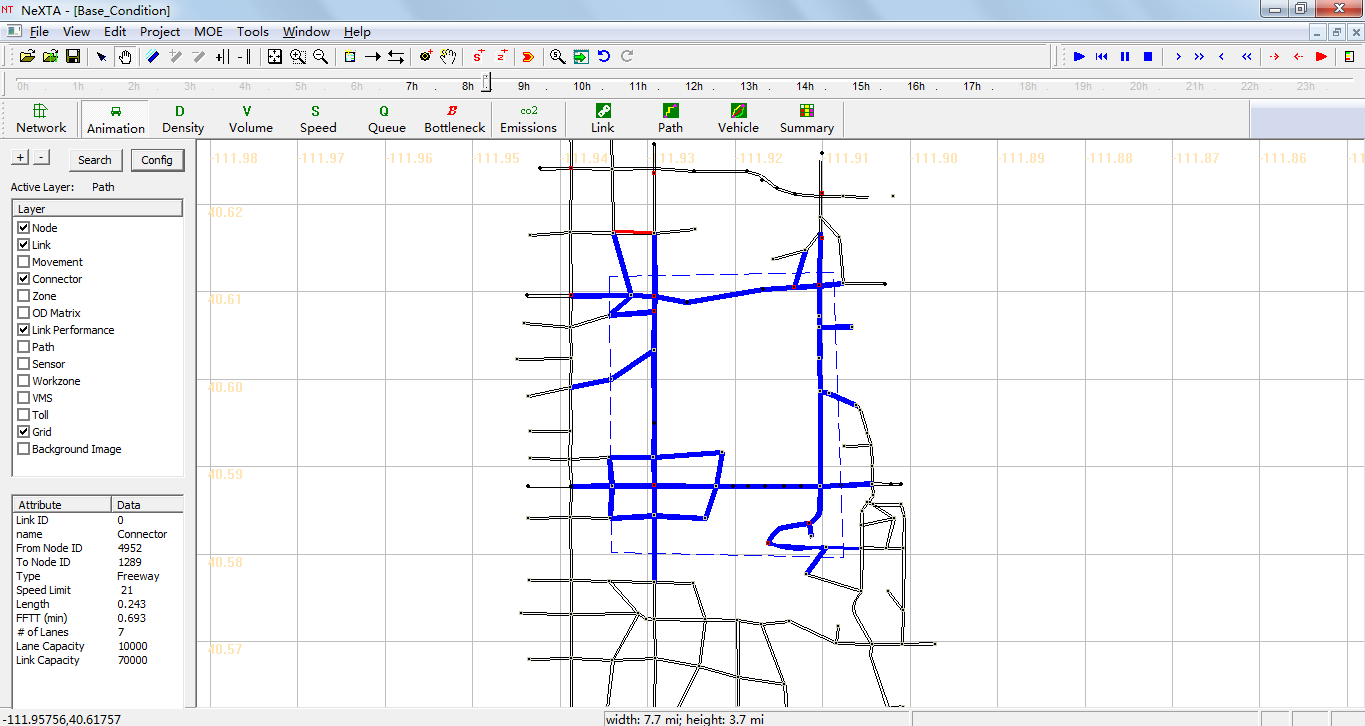




The third drop-down list controls the area or features over which statistics are collected in the chart. The default is for network-wide statistics, but the user may also choose a set of OD Pairs, a Path (using the method described for using the Path Selection Tool), a link or set of links.



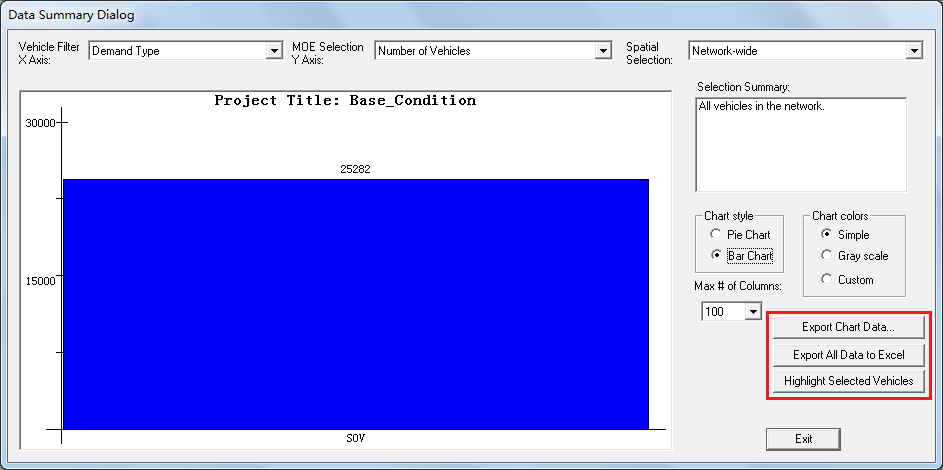
Using the Subarea Selection Tool, the user can also choose from a set of subarea-related options, including analyzing vehicles leaving the selected subarea, vehicles traversing the subarea, subarea trips disaggregated by internal-external, external-internal, and internal-internal trips, complete and partial trips inside the subarea and subarea boundary-to-boundary trips. The user can create a subarea by selecting the  button on the upper Toolbar, or by going to Edit > Create Subarea.



After initiating the tool, the user can create a subarea in the larger network by selecting points in a polygon which defines the subarea. The subarea is completed by clicking very close to the location of the first point which defines the subarea. After the subarea is defined, the subarea is represented with a dotted white line, and the links contained within the subarea are selected (highlighted in blue), as shown in the figure above.

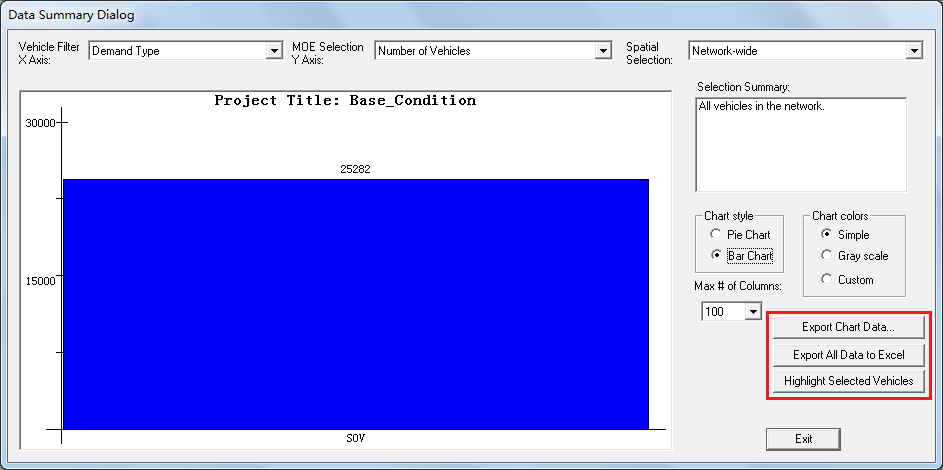


Visualization options are available at the right side of the Data Summary Dialog window, with Chart style (Pie or Bar Charts) and Chart colors available for modification (as shown in the figure above). The Simple color option uses a default color selection, and the Gray scale color option performs a simple conversion for black and white diagrams. The Custom color option uses different cross-hatching patterns for visualization.



Three buttons are available at the bottom right side of the window – Export Chat Data, Export All Data to Excel and Highlight Selected Vehicles (as shown in the figure above).

Clicking the Export Chart Data button will export the data used to create the chart currently displayed in the right panel. Selecting the Export All Data to Excel will export all data and save them as CSV files. The Highlight Selected Vehicles button will make relevant link colors green to indicate where the selected vehicles travel in the network. After pressing the button, pressing the button again will hide the highlighted vehicles (remove blue link highlighting).



# SCENARIO TESTING

## Basic Traffic Assignment Scenario

The basic data input for a dynamic network loading program (f) includes time-dependent origin-destination demand (a) and the traffic network (b) with road capacity constraints on links and nodes. However, it is a challenge to obtain accurate dynamic demand and real-world network data required for practical application.

Through calculating the dynamic travel time (c) of all links (free-flow travel time is used at first iteration), the route choice model (d) embeds a standard time-dependent least-cost path algorithm (e) to generate paths for all agents. The combination between least cost and the behavior of traveler’s choice is also still one challenging issue.

The core dynamic network loading program (f) loads previously generated agents on the traffic network for the entire planning horizon, which produces link-based traffic states (h) that describes time-varying traffic states at the link level using traffic flow model (g) and updates the dynamic travel time database (c).

At the following step within an iterative assignment process, the route choice module (d) will again re-compute the route selection for each agent using updated travel times (c), for another iteration of dynamic network loading (f), until the model is converged or reaches the maximum number of iterations.

#### Step 1: Import Network from GIS Shape Files

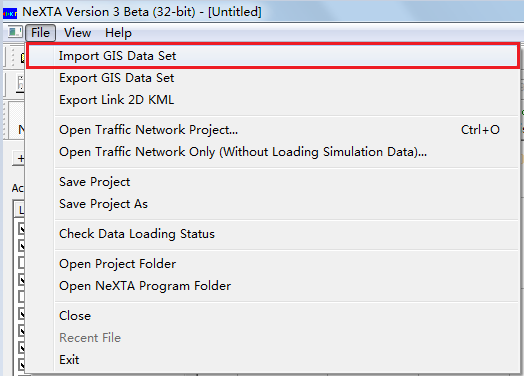
* Step 1-1: Prepare GIS shape files, and save to the destination folder
* Open network files in Cube(or ArcGIS)
* Change network to WGS 84 coordinate system
* Check the available data. It should be noted that necessary attribute data of links and nodes should be prepared for network import, an example is shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Attribute** | **Remark** | **Field Name in Example \*.shp Files** |
| Node | Node ID | Required | NODE\_ID |
| Name | Desired | NODE\_NAME |
| TAZ | Desired |  |
| Control Type | Desired | NODE\_TYPE |
| Link | From Node ID | Required | A |
| To Node ID | Required | B |
| Name | Desired | LINK\_NAME |
| Link ID | Required | LINK\_ID |
| Link Type | Desired | LINK\_TYPE |
| Direction | Desired | DIR |
| length | Required | LENGTH |
| Number of lanes | Required | LANE |
| Hourly Capacity | Required | CAP |
| Speed Limit | Required | SPEED |
| r\_number\_of\_lanes |  |  |
| r\_hourly\_capacity |  |  |
| r\_speed\_limit |  |  |
| r\_link\_type |  |  |

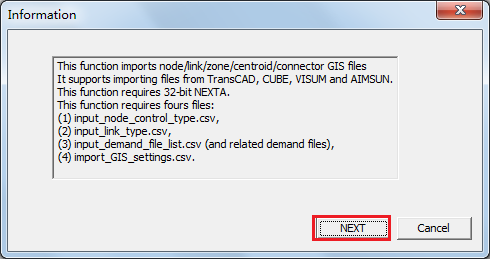
* Export GIS shape files
* Export node shape file and link shape file when provided
* The corresponding files will be written to the destination folder
* Step 1-2: Prepare file for importing GIS network
* Copyfile from any sample data set to the destination folder
* Setto read the corresponding GIS shape files
* Value of “link\_ file\_ name” = link shape file’s name
* Value of “node\_ file\_ name” = node shape file’s name
* In order to read specific data in \*.shp files, set the value in node/link section to corresponding file name in \*.shp files, as shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Key** | **Remark** | **Value according to example \*.shp files** |
| Node | node\_id |  | NODE\_ID |
| name |  | NODE\_NAME |
| TAZ |  |  |
| control\_type | Necessary when user have signal control type | NODE\_TYPE |
| Link | from\_node\_id |  | A |
| to\_node\_id |  | B |
| name |  | LINK\_NAME |
| link\_id |  | LINK\_ID |
| link\_type | default value = 1 | LINK\_TYPE |
| mode\_code | reserved for transit modeling |  |
| direction | “1”or “2” | DIR |
| length |  | LENGTH |
| number\_of\_lanes |  | LANE |
| hourly\_capacity |  | CAP |
| speed\_limit |  | SPEED |
| r\_number\_of\_lanes |  |  |
| r\_hourly\_capacity |  |  |
| r\_speed\_limit |  |  |

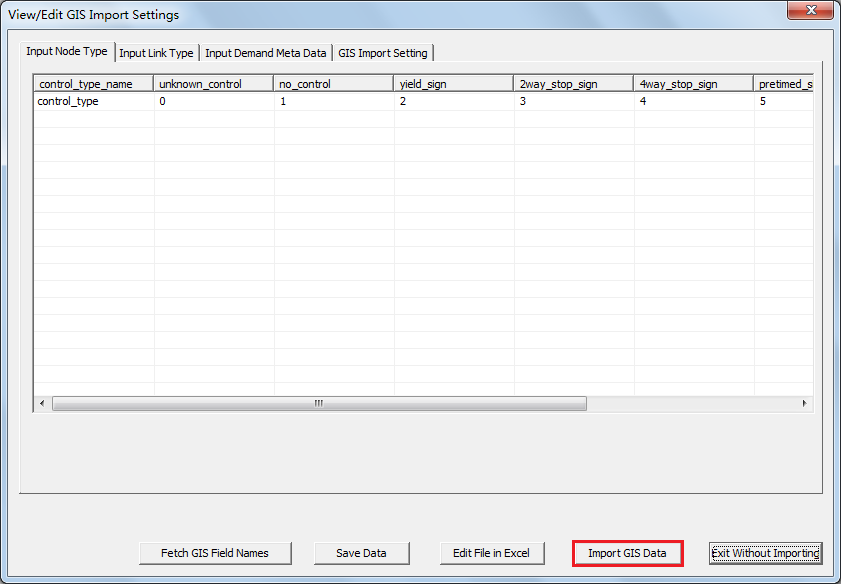
* Step 1-3: Open  , and import GIS network
* File> > Import GIS Data Set



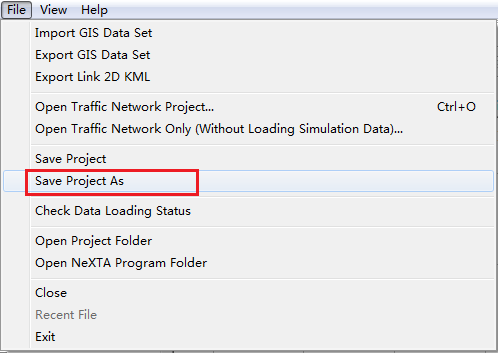
* Click button “Next”



* Choosefile from the destination folder > open
* Click button “Import GIS Data”



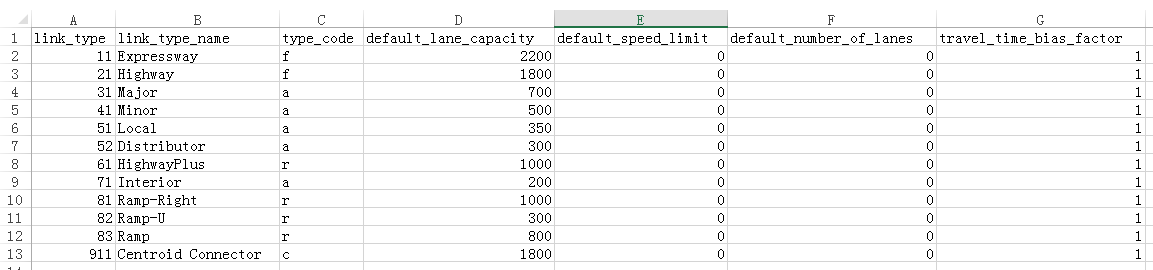
* Step 1-4: Save network as a \*.tnp file to the destination folder
* File> Save Project As



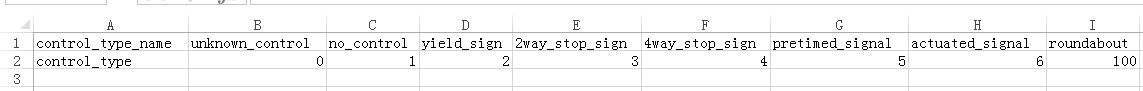
* Save network as a \*.tnp file to the destination folder> Click button “Ok”

#### Step 2: Prepare Necessary CSV Files

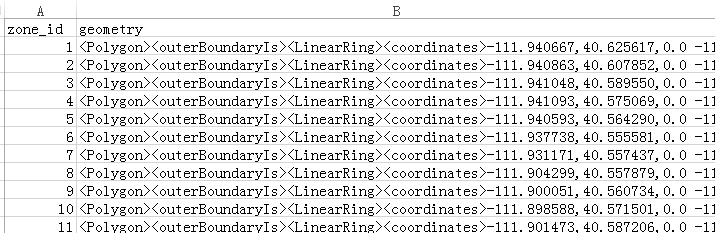
* Step 2-1: Preparefile
* Copyfile from any sample data set to the destination folder
* Setif necessary, as well as default settings



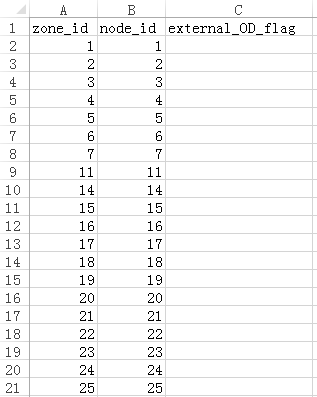
* Step 2-2: Preparefile
* Copyfile from any sample data set to the destination folder
* Setif necessary, as well as default settings



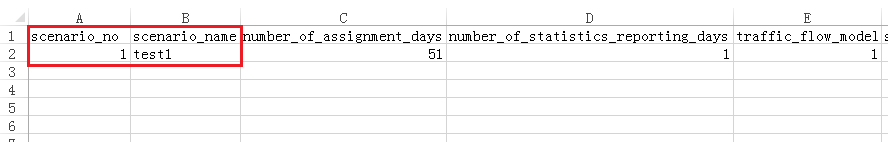
* Step 2-3: Prepare file
* Copyfile from any sample data set to the destination folder
* Specify zone ID
* The value of geometry in this file is not necessary, which can be specified in file or file



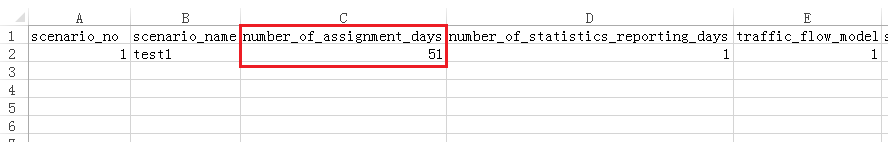
* Step 2-4: Prepare file
* Copyfile from any sample data set to the destination folder
* Specify zone id (correspond to zone id in file) and the corresponding node id (correspond to node id in file)
* Specify external\_OD\_flag= 1 for external zones (zero for none)



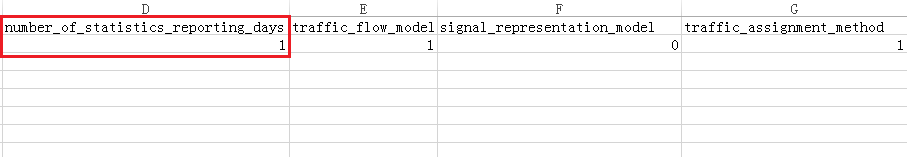
* Step 2-5: Prepare files
* Copyfile from any sample data set to the destination folder
* Specify the sequence number and the corresponding name of scenarios



* Specify the number of iteration (e.g., 51 as shown in the example file)

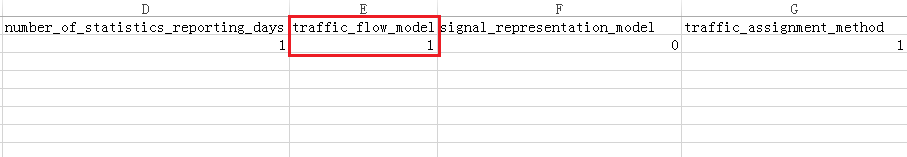


* Specify the number of statistics reporting days



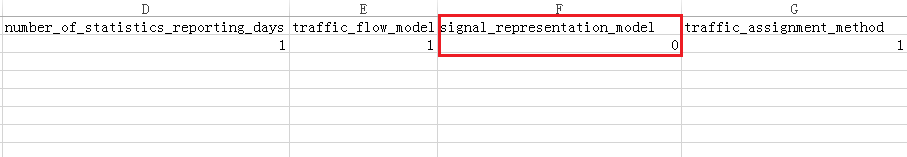
* Specify traffic flow model

|  |  |
| --- | --- |
| No. | Traffic Flow Model |
| 0 | BPR Function |
| 1 | Point Queue Model |
| 2 | Spatial Queue Model |
| 3 | Newell’s Kinematic Wave Model |
| 4 | Newell’s Model Emissions Output |
| 5 | User Define Traffic Flow Model |



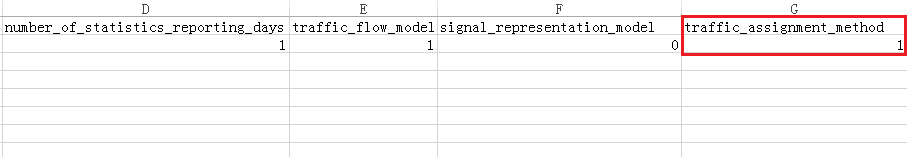
* Specify signal representation model

|  |  |
| --- | --- |
| No. | Signal Control Representation |
| 0 | Continuous Flow with Link Capacity Constraint |
| 1 | Cycle Length+ Link-based Effective Green Time |
| 2 | Phase Based Signal Representation Model |

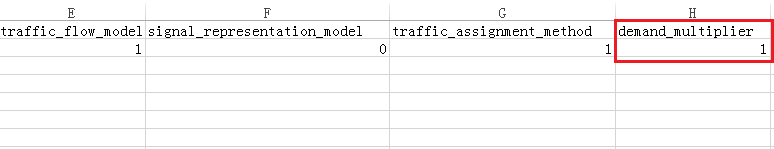


* Specify traffic assignment method

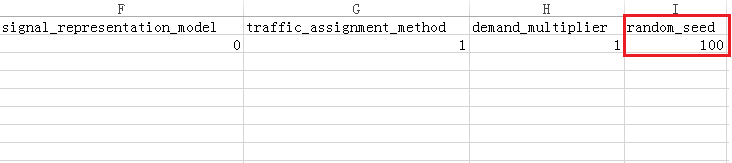
|  |  |
| --- | --- |
| No. | Traffic Assignment Method |
| 0 | Method of Successive Average |
| 1 | Fixed Switching Rate |
| 2 | Day-to-Day Specific Learning Rate |
| 3 | OD Demand Matric Estimation |
| 4 | Day-to-Day Route/Departure Time Choice with BR rule |
| 5 | Accessibility(Travel Time) |
| 6 | Accessibility(Distance) |
| 7 | Scenario Evaluation Based on Agent Binary File |
| 8 | Real Time Simulation Base on External Data EXchange |
| 9 | Real Time Simulation with Activity Based Model |
| 10 | System Optimal Based on Agent Binary File |



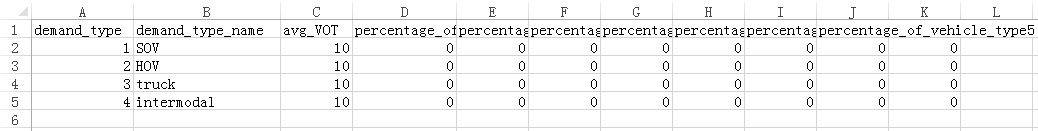
* Specify the multiplier of loading demand for given scenario



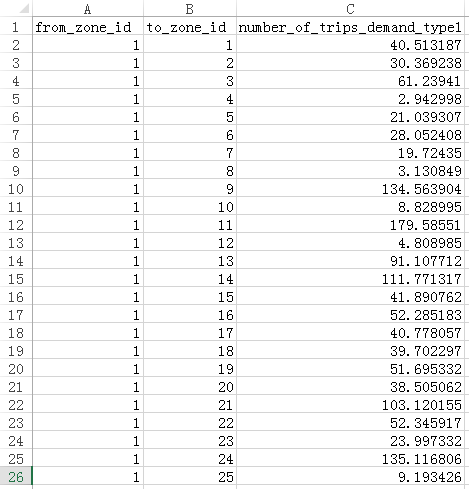
* Specify the seed number used for the pseudorandom number generator

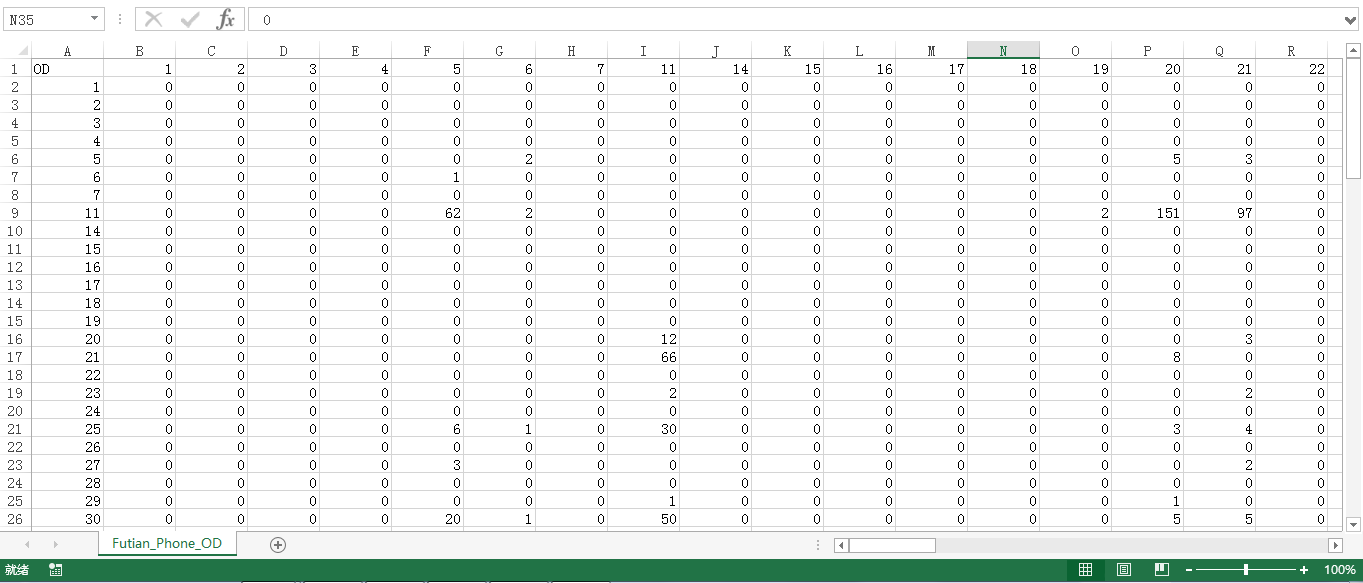


* Step 2-6: Preparefile
* Copyfile from any sample data set to the destination folder
* Setif necessary, as well as default settings

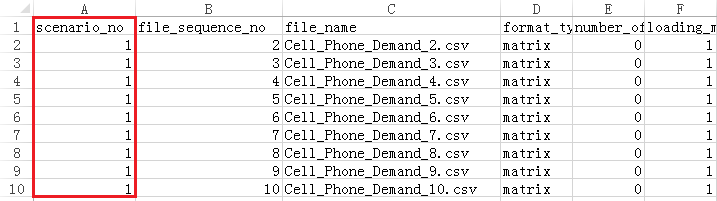


* Prepare input demand data file
* Prepare demand data files and save as \*.csv files to the destination folder
* The format type of demand data (e.g., column or matrix , as shown in the figure blew), see more detail in section 2.3.3. Simulation Configuration Files.

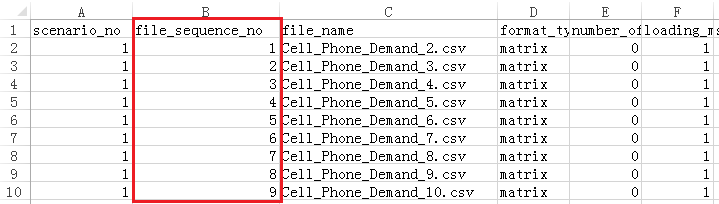




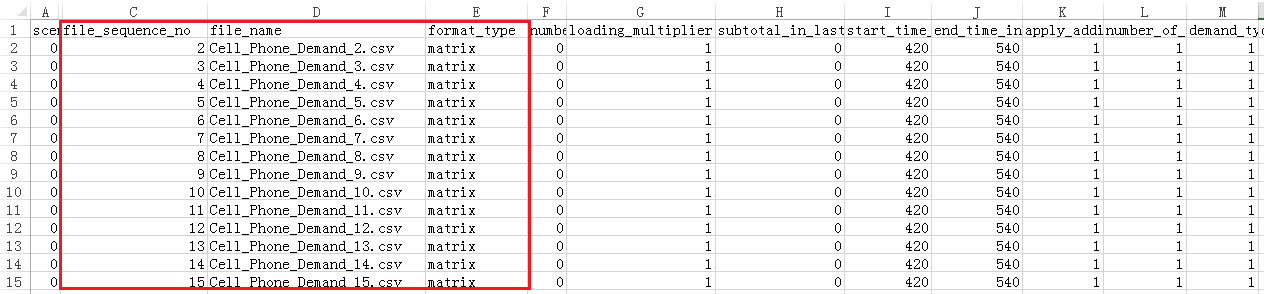
* Step 2-7: Preparefile
* Copyfile from any sample data set to the destination folder
* Setfile to read the corresponding input demand data files
* Specify scenario\_no (correspond to the value of “scenario\_no” infile)



* Specify the sequence number of demand data files (e.g., 1, 2, 3…)



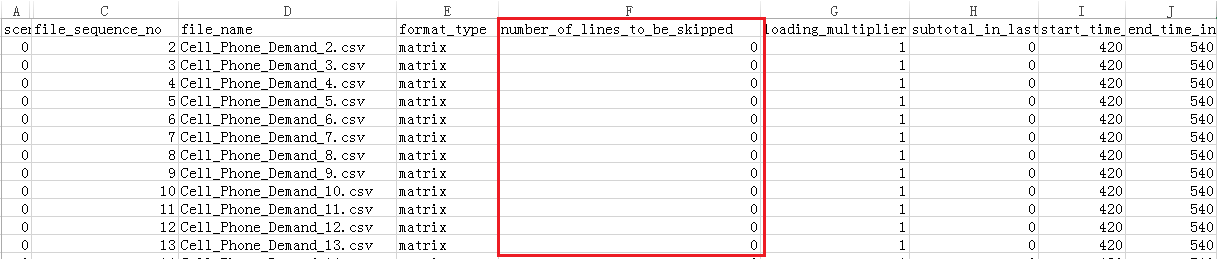
* Specify the demand file name (e.g., Cell\_Phone\_Demand\_2.csv) and format (column, or matrix)



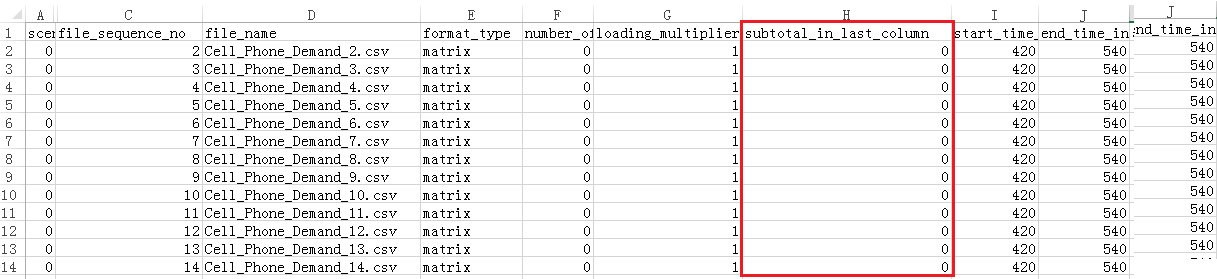
* Specify multiplier of loading in the demand file (e.g., 1)



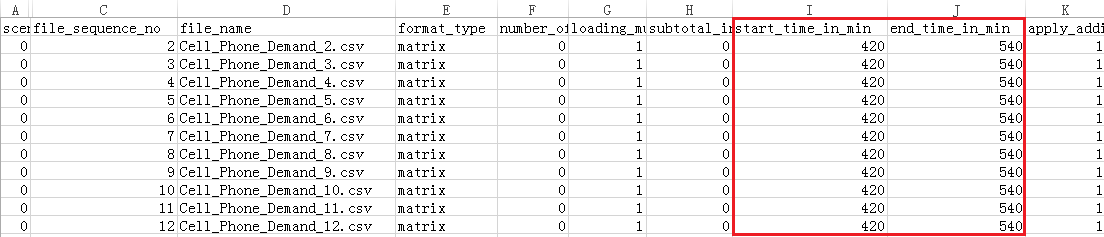
* Specify the number of lines in the demand file to be skipped by NeXTA (8 for MTX file)



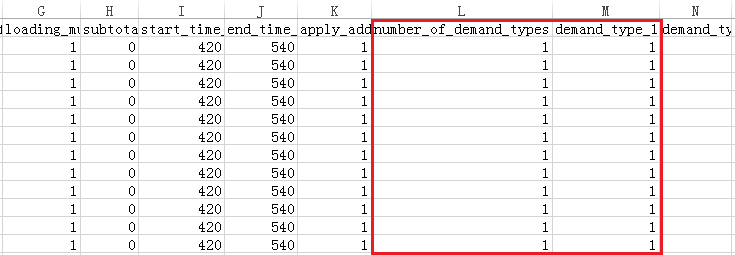
* Indicate whether subtotals are present in the last column (zero for none)



* Specify the loading start time and end time for the demand file (420 to 540, or 7AM to 9AM)



* Specify the demand types associated with the demand file (e.g., Only demand type 1)



#### Step 3: Run Simulation and Perform Analysis

* Step 3-1: Check imported network (coordinate system, link, node, demand...) and imported CSV files
* Step 3-2: Double-click , run simulation and perform analysis

## Toll Scenario

In DTALite, an agent/driver chooses a route based on the generalized cost to travel between an origin and destination. This generalized user cost is based upon three components: travel time, value of time (VOT), and tolling/pricing: Cost = Travel Time×VOT + Toll

Each variable in the generalized cost equation is specific to the agent/driver. Every agent/driver has their own value of time, travel times through the network are estimates which are dependent upon the type of information available to that agent/driver, and the toll is dependent upon the demand or vehicle type (SOV vs. HOV).

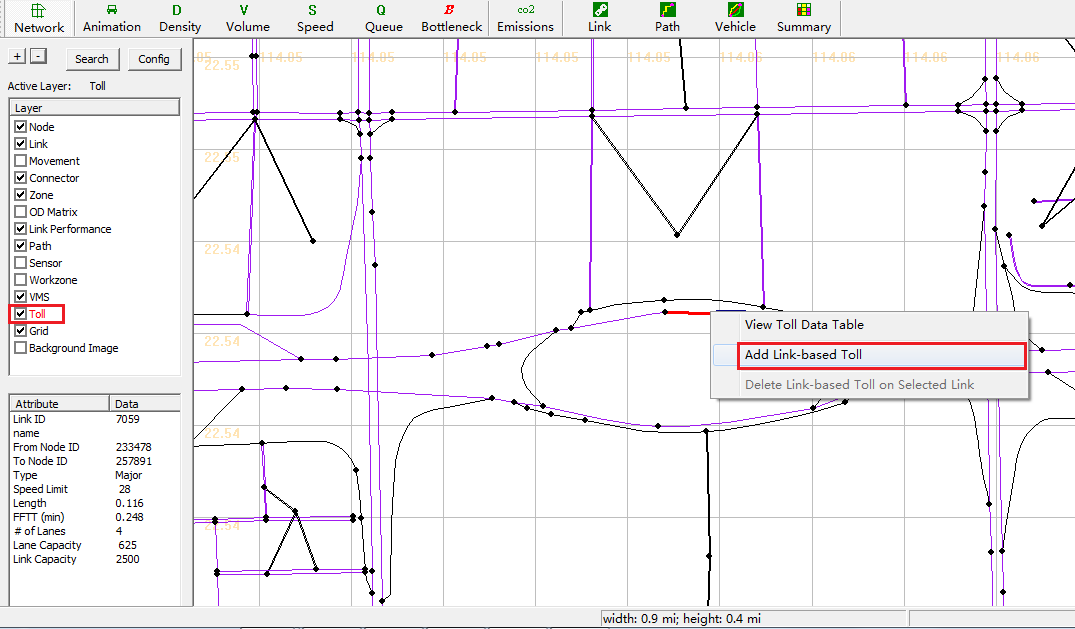
When offered multiple paths with different estimated travel costs, the agent/driver is assumed to select the path/route which minimizes their own cost (Wardrop’s 1st Principle). When making successive decisions about route choice (e.g., choosing a new path after departure, and/or choosing a path in the second simulation iteration), the new path must overcome the “willingness to switch” paths, which is a relative threshold used when comparing travel times. That is, the agent/driver may decline to change paths if the alternative path does not meet some minimum amount of travel time (i.e., 5% travel time savings).

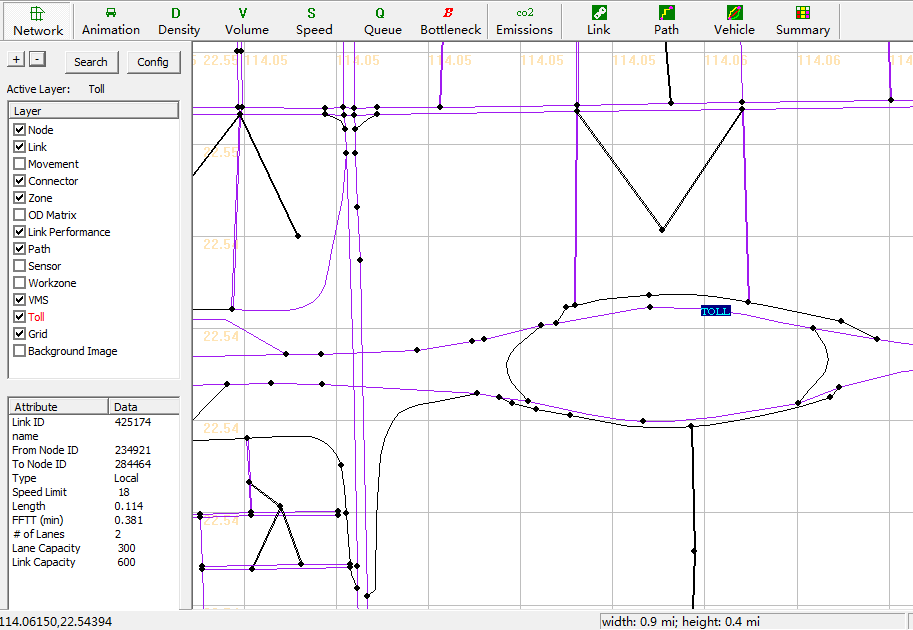
#### Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

See more details in **section 4.1. Basic Traffic Assignment Scenario**.

#### Step 2: Add Toll on Select Links in NeXTA

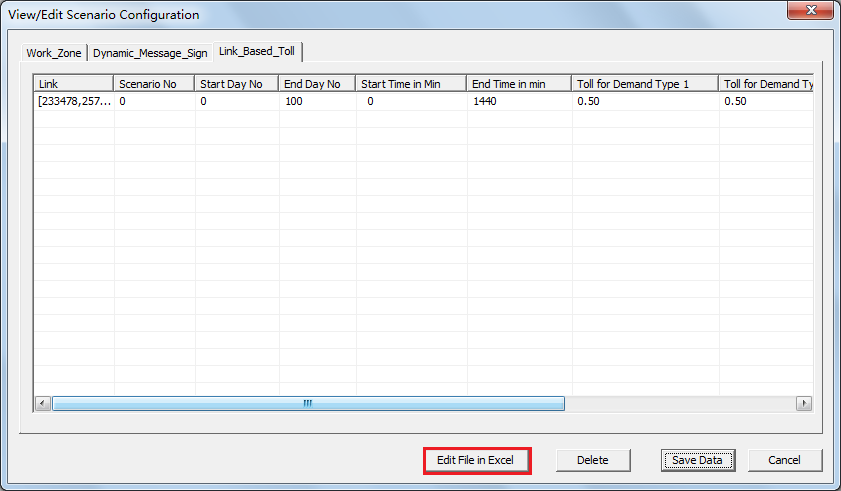
* Step 2-1: Select toll layer, use the select object tool  to select a link
* attributes of selected link could be seen in the left side below the screen, including origin node ID, destination node ID, link type, link capacity…
* Step 2-2: Right-click the selected link and select the menu “Add Link-based Toll”



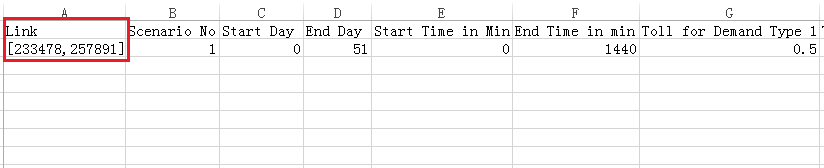


#### Step 3: Prepare Specific Input Files for Toll Scenario

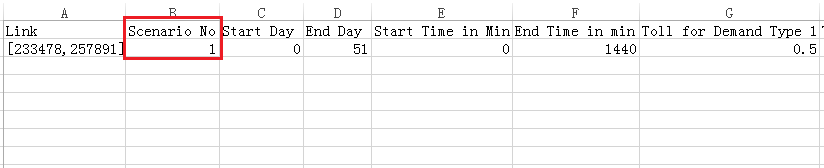
* Step3-1: Set  file
* Click button “Edit File in Excel” and open file



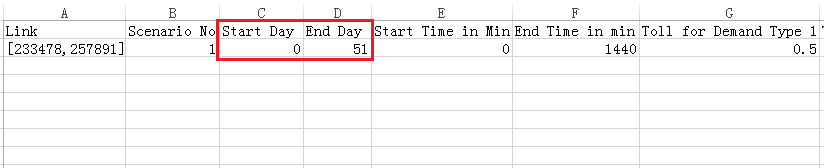
* Check origin node and destination node of Selected links



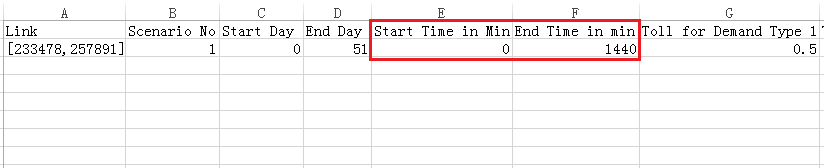
* Specify the sequence number of scenario (according to the value of “Scenario\_No” in file)



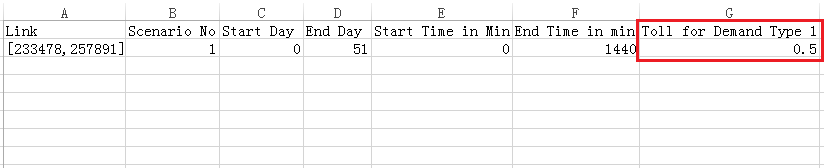
* Specify the iteration of toll scenario



* Set the star time and end time of toll scenario in specified iteration



* Set toll standard of different demand types



* It should be noted that, different charging standard could be set to affect travelers’ route choice. For example,

#### Step 4: Run Simulation and Perform Analysis

* Step 4-1: Double-click  after checking all input files, run simulation and perform analysis

## Emissions Scenario

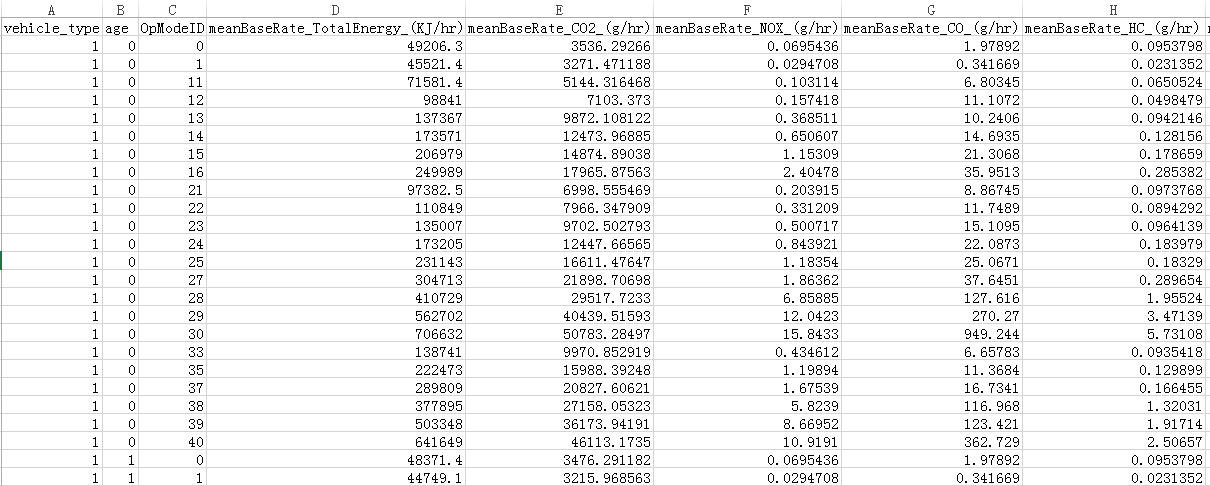
For estimating vehicle emissions in DTALite, we need to prepare the necessary file(as well as default) that defines emissions rate for a variety of vehicle types, and calculate the total emissions of vehicles in the network.

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

See more details in section **4.1. Basic Traffic Assignment Scenario**.

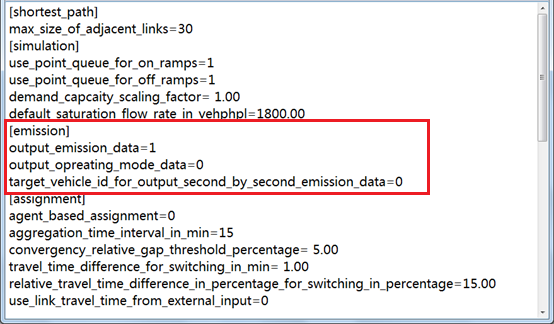
#### Step 2: Prepare Specific Input Files for Outputting Emissions Data

* Step 2-1: Setfile
* Copyfile from any sample data set to the destination folder
* Set if necessary, as well as default settings



#### Step 3: Run Simulation and Perform Analysis

* Step 3-1: Set  file for outputting emission data



* Step 3-2: Double-click  after checking all input files, run simulation and perform analysis

## Work Zone Scenario

Work Zone Scenario could be implemented in the following conditions:

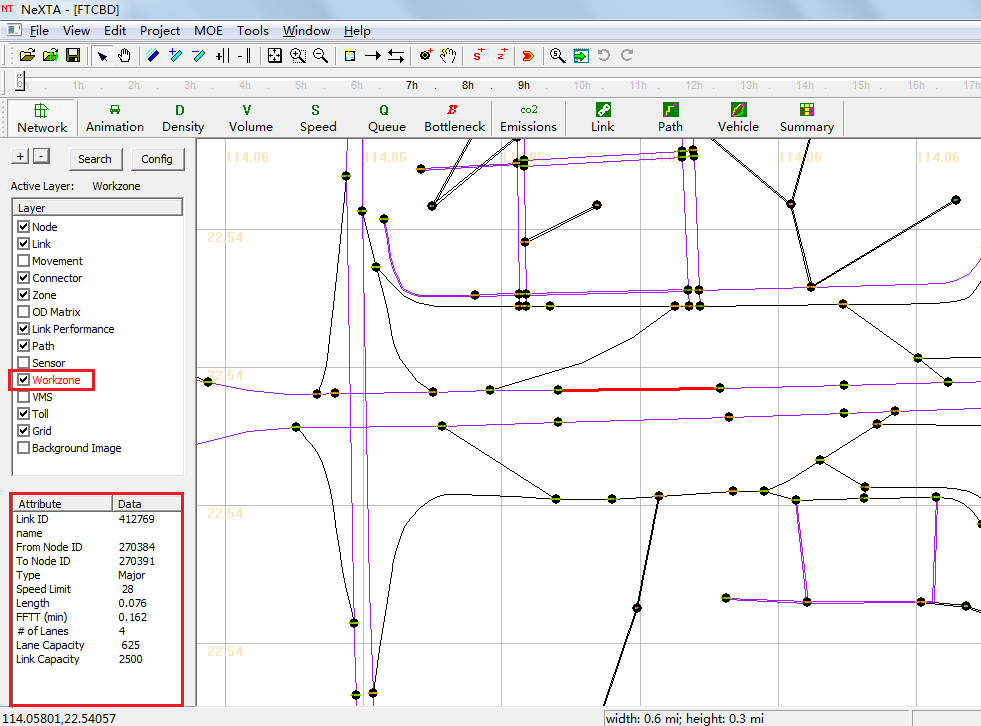
1. Capacity reduction or even blockades on the roads for some time caused by traffic incident, work zone and bad weather
2. Regular traffic management or special traffic control for special events
3. Parking control management

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

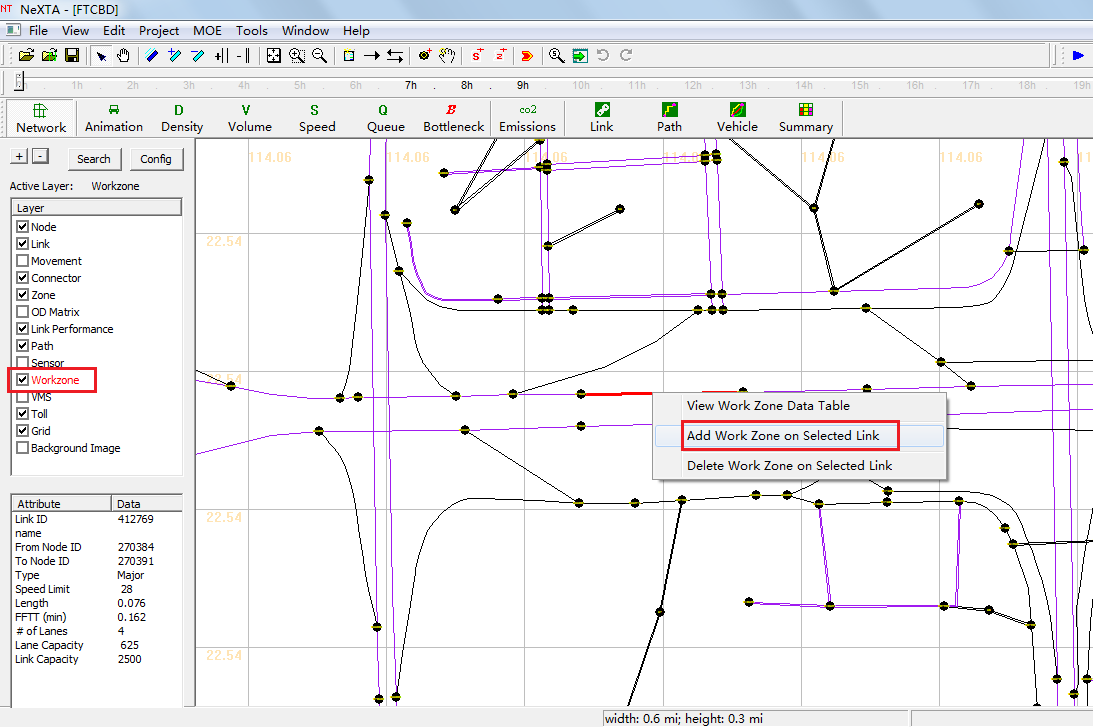
See more details in section **4.1. Basic Traffic Assignment Scenario**.

#### Step 2: Add WorkZone on Select Links in NeXTA

* Step 2-1: Select workzone layer, use the select object tool  to select a link
* attributes of selected link could be seen in the left side below the screen, including origin node ID, destination node ID, link type, link capacity…

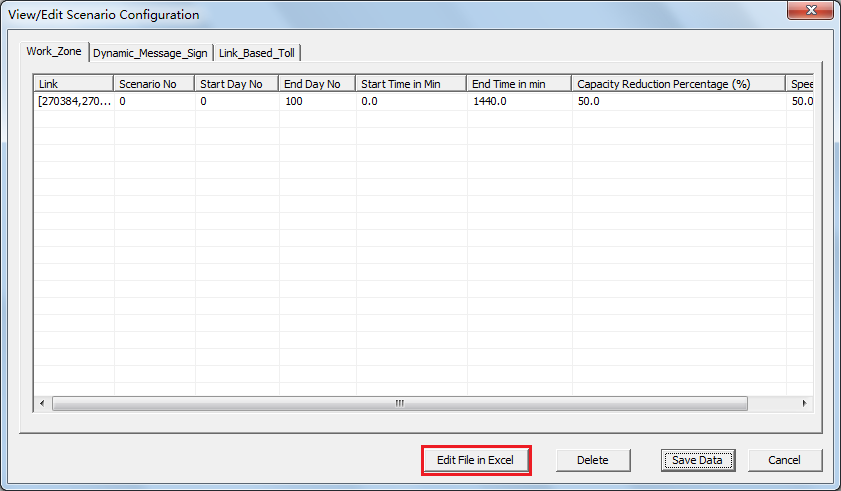


* Step 2-2: Right-click the link and select the menu “Add Work Zone on Selected Link”

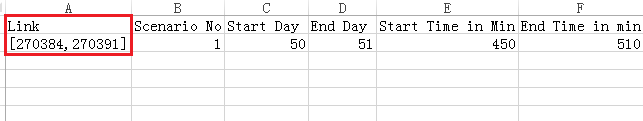


#### Step 3: Prepare Specific Input Files for WorkZone Scenario

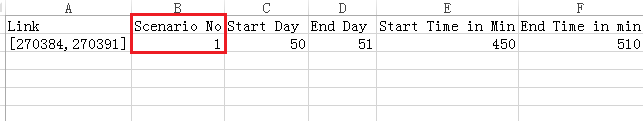
* Step3-1: Set  file
* Click button “Edit File in Excel” and open file



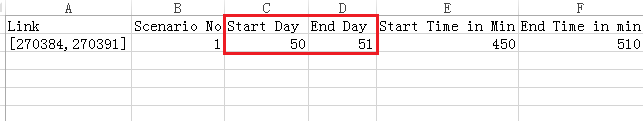
* Check origin node and destination node of Selected links



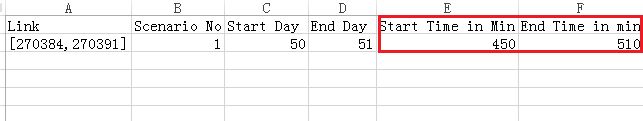
* Specify the sequence number of scenario (according to the value of “Scenario\_No” in file)



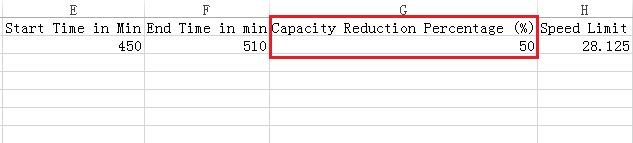
* Specify the iteration of work zone scenario
* In this example, the number of simulation iteration is set to “51” in file (more details in section **4.1. Basic Traffic Assignment Scenario**)
* The start day and end day of simulation iteration should be set to a large enough number to make sure work zone scenario is implemented under such conditions that dynamic traffic assignment and agents’ path choice in the whole network is stable.(e.g., “Start Day”=50”and “End Day”=51 in this case)



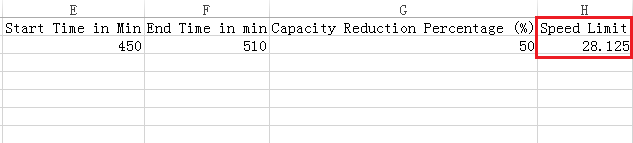
* Set the star time and end time of work zone scenario in specified iteration
* It should be noted that, work zone scenario should start after demand loading for a period of time.( According to  file)



* Specify capacity reduction for work zone scenario
* If the capacity reduction percentage is set to 100%, then no vehicles could use that link.

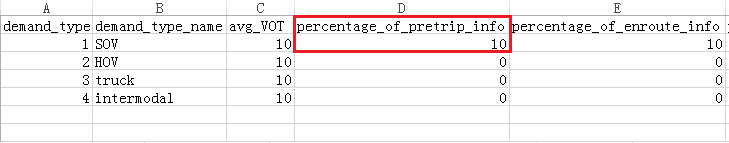


* Specify the value of limit speed of selected links for work zone scenario

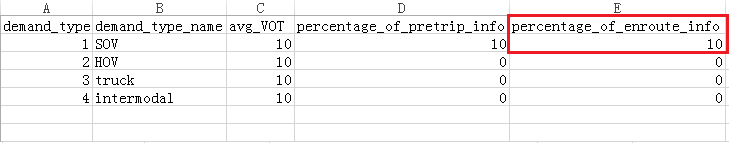


#### Step 4: Additional Information Setting (Pre-trip & En-route)

* This step should be carried out when there are additional information for travelers such as pre-trip and en-route.
* Open file to provide additional information to travels of different demand types.
* Specify percentage of pre-trip information for travelers of different demand types



* Set ratio of en-route information for travelers from different demand type



#### Step 5: Run Simulation and Perform Analysis

* Step 5-1: Double-click  after checking all input files, run simulation and perform analysis

## VMS Scenario

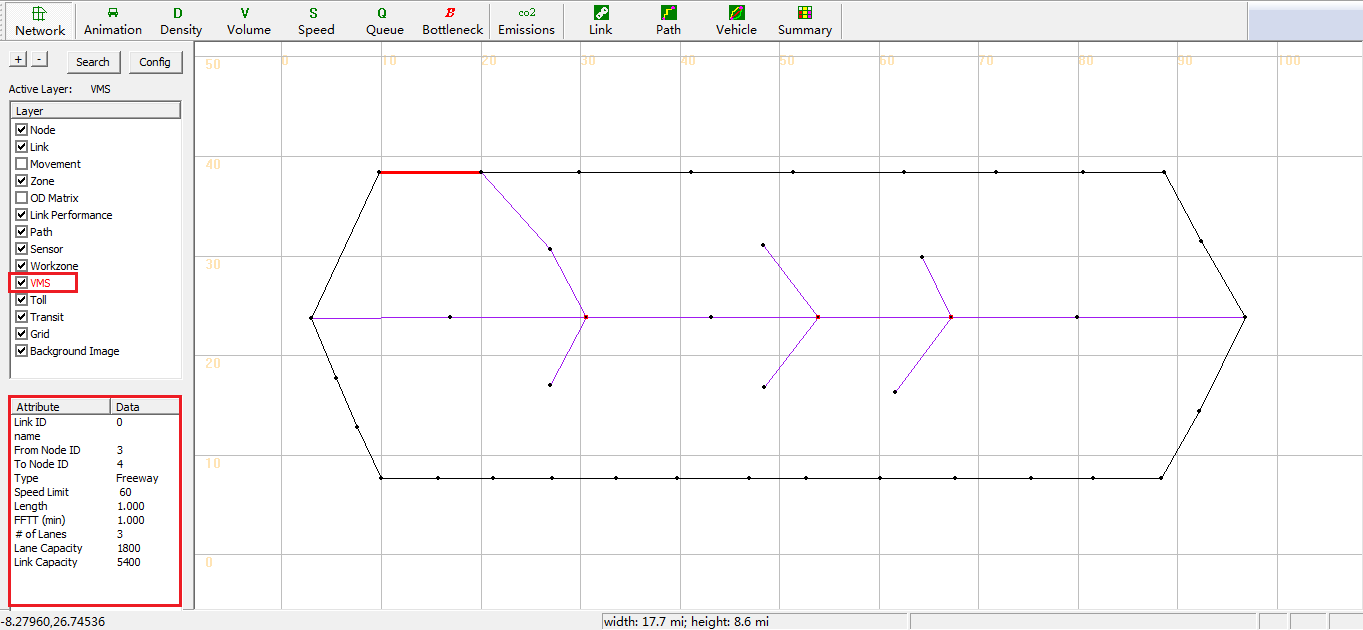
Variable Message Signs (or Dynamic Message Signs) are used in DTALite to cause travelers to consider changing their current route. A VMS is located on a link in the network, and vehicles passing through the link are potentially affected by the sign. Each VMS has a response rate specified by the user, representing the probabilistic percentage of travelers which are affected by the sign and decide to consider a new route to their destination.

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

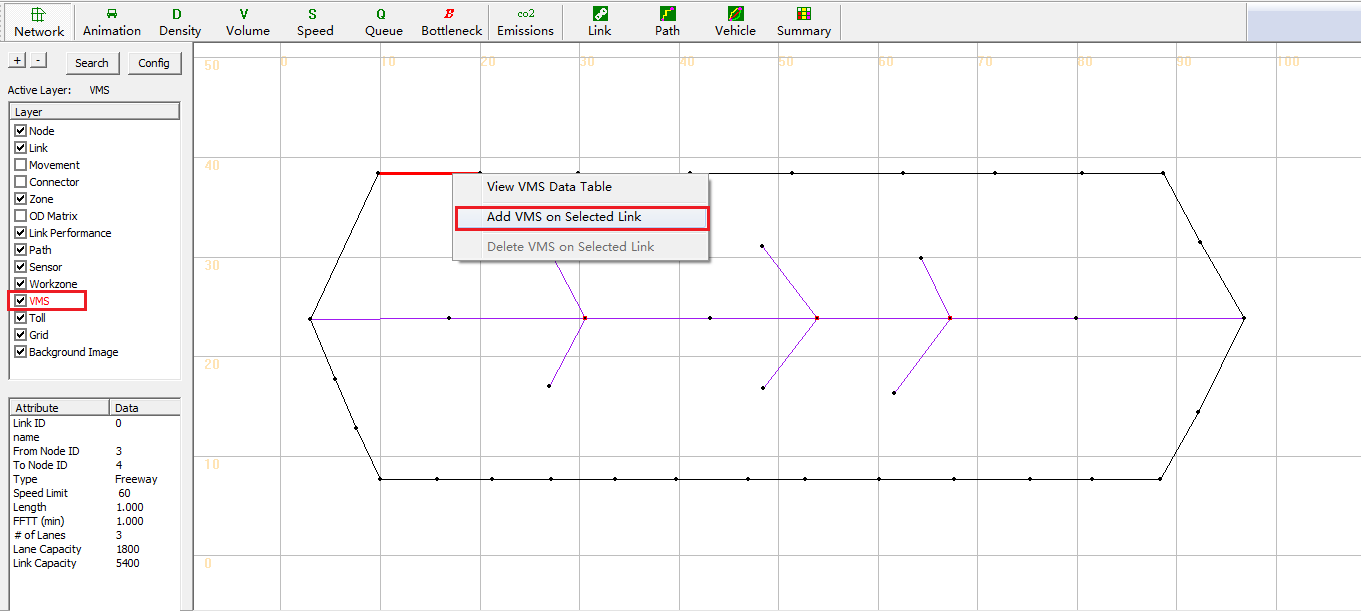
See more details in section **4.1. Basic Traffic Assignment Scenario**.

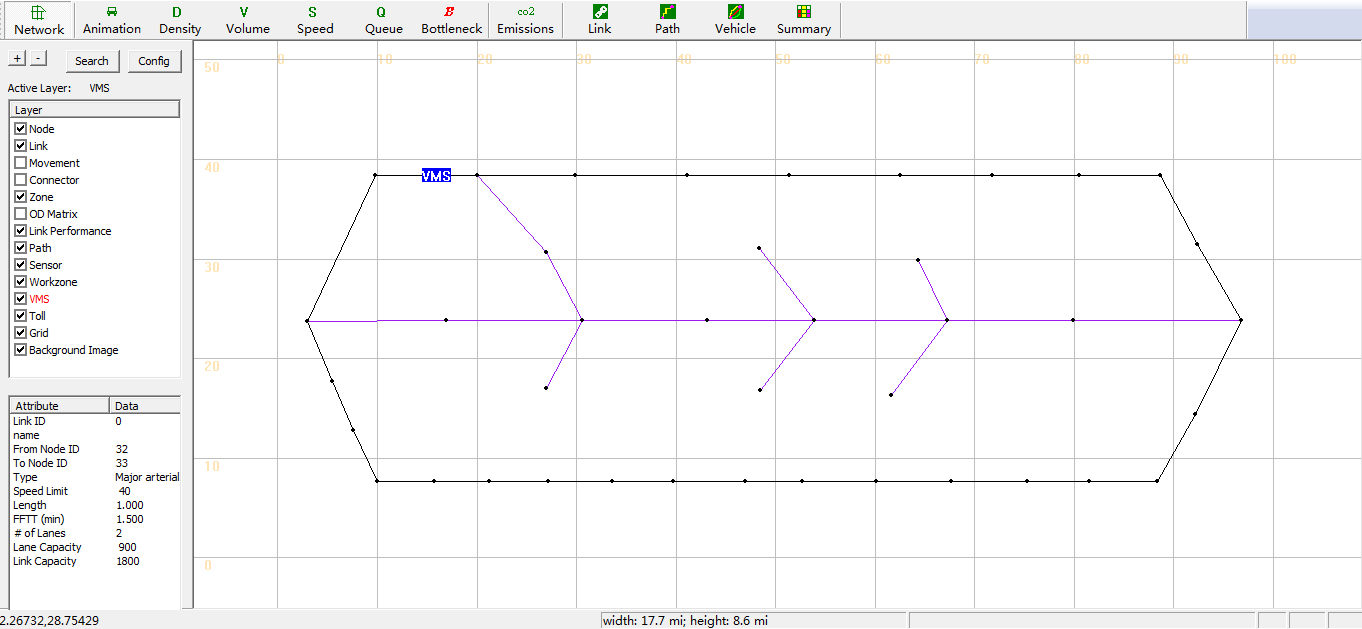
#### Step 2: Add VMS on Select Links in NeXTA

* Step 2-1: Select the VMS layer, use the select object tool  to select a link
* attributes of selected link could be seen in the left side below the screen, including origin node ID, destination node ID, link type, link capacity…



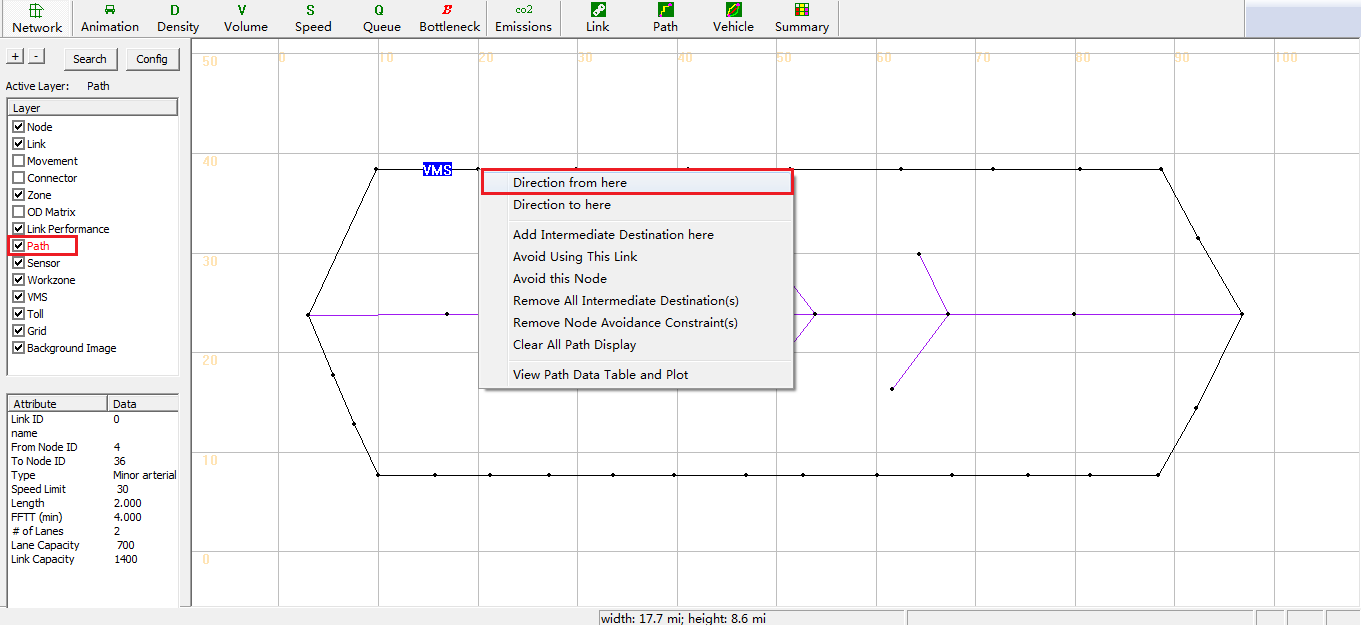
* Step 2-2: Right-click the link and select the menu “Add VMS on Selected Link”



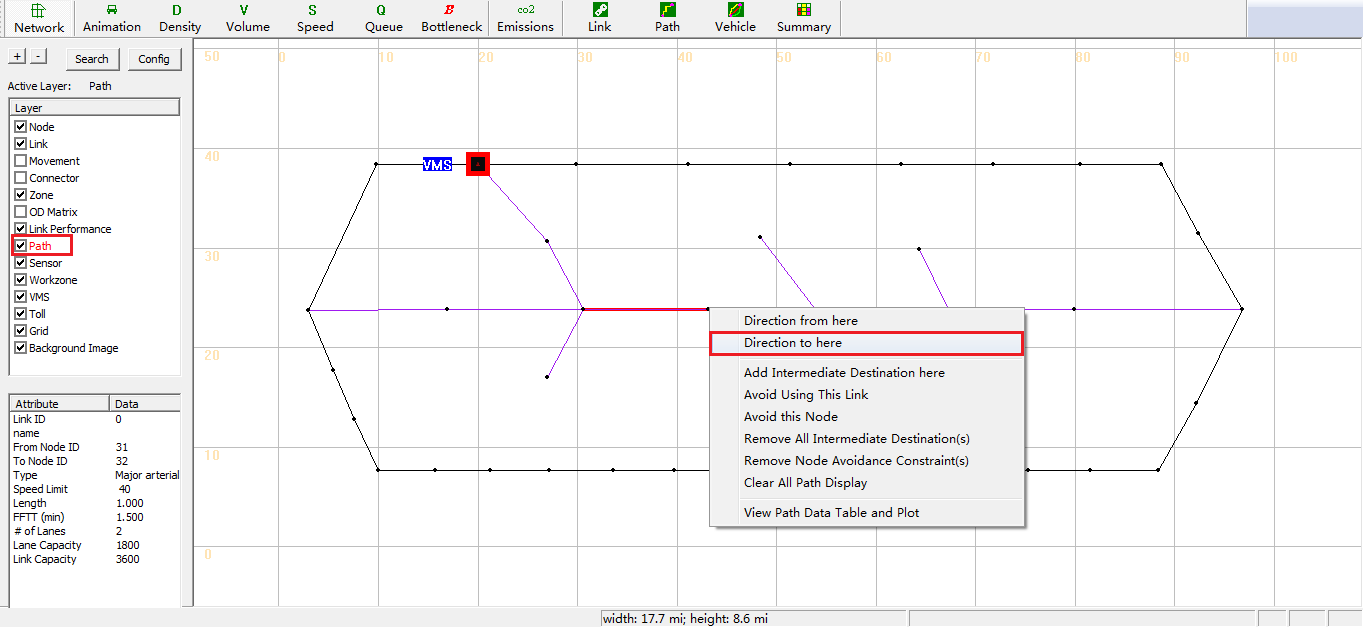


#### Step 3: Gain necessary information of detour routes

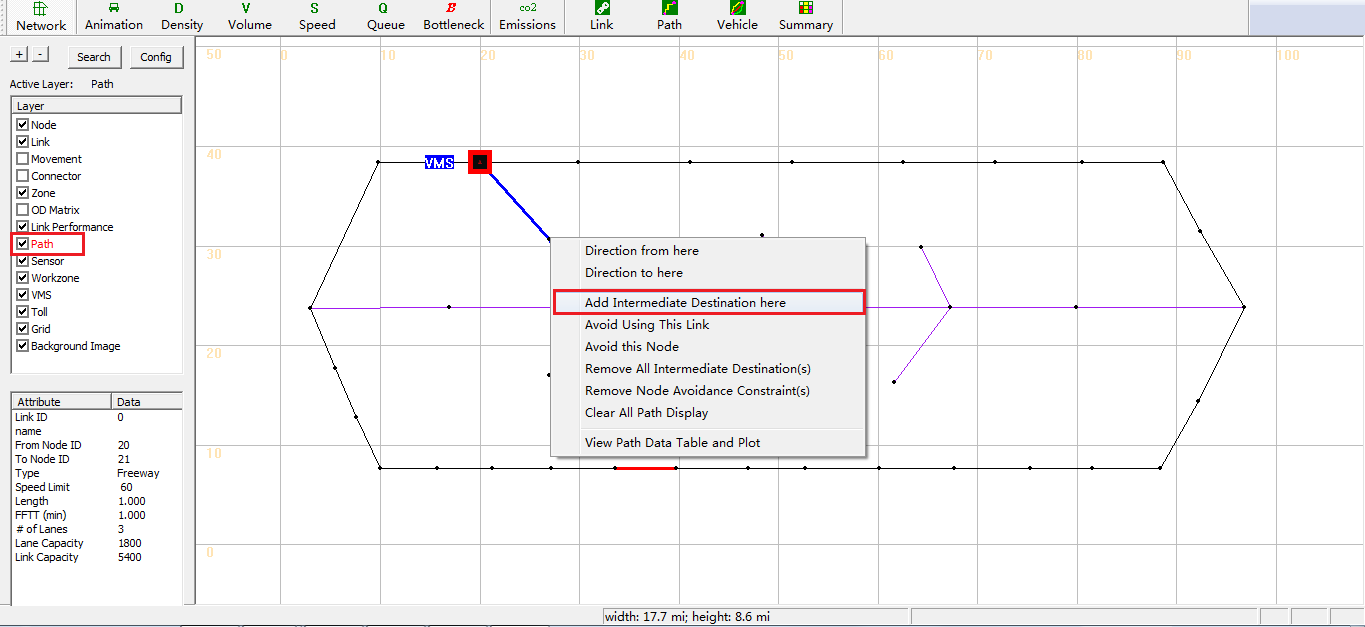
* Step3-1: Select path layer, use the select object tool  to select a path
* Select the destination node of VMS link as the origin node of detour route, right-click the node and select menu “Direction from here”. Because VMS is only effective when placed upstream from a detour point.



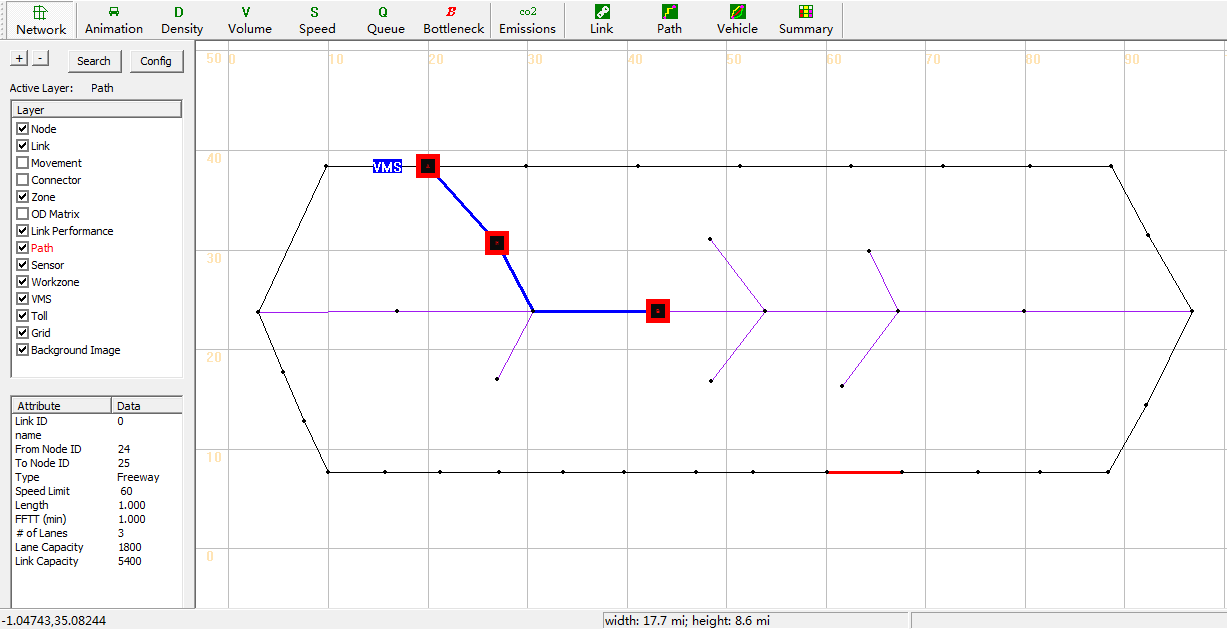
* Select the destination node of detour route, right-click the node and select menu “Direction to here”



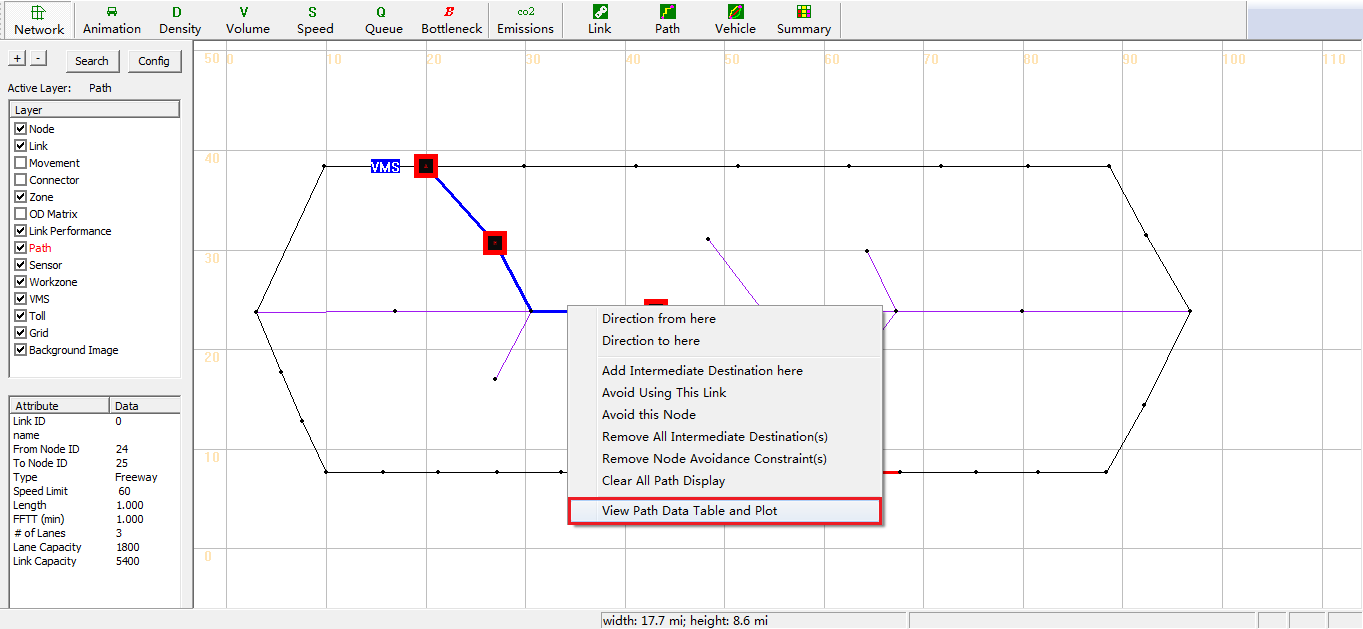
* Select the intermediate node of detour route if necessary, right-click the node and select menu “Add intermediate Destination here”



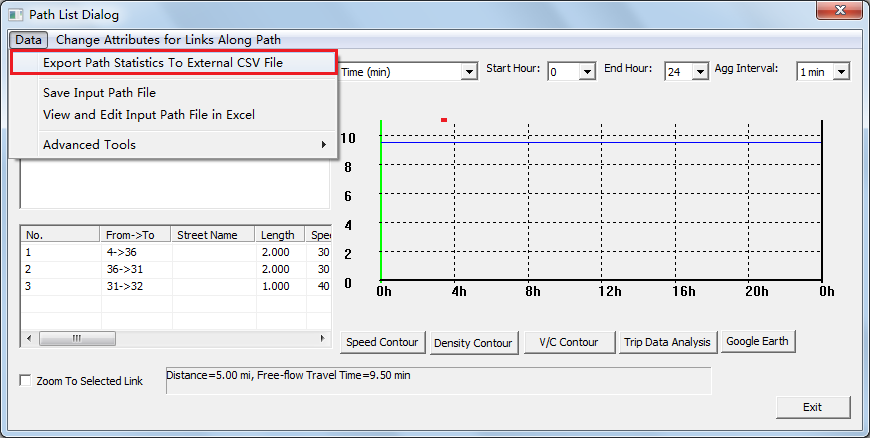
* Then a detour route is selected as shown the blue path in the figure below.



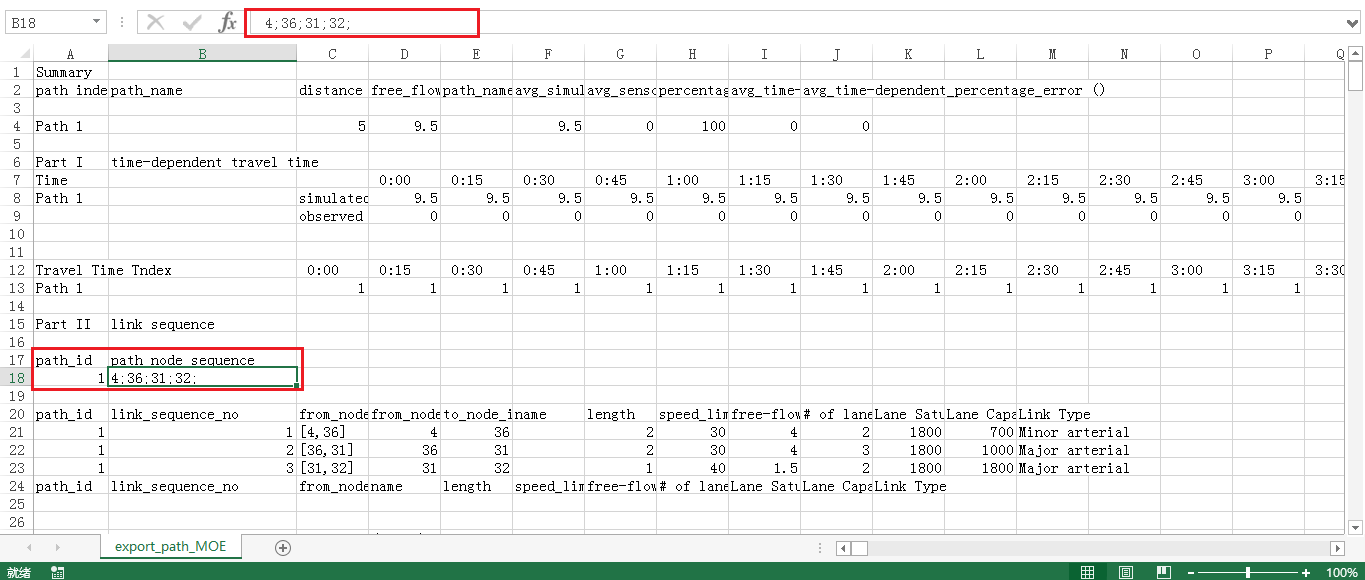
* Step3-2: Right-click the selected path and choose the menu “View Path Data Table and Plot”



* Step3-3: Click Data>Export Path Statistics To External CSV File

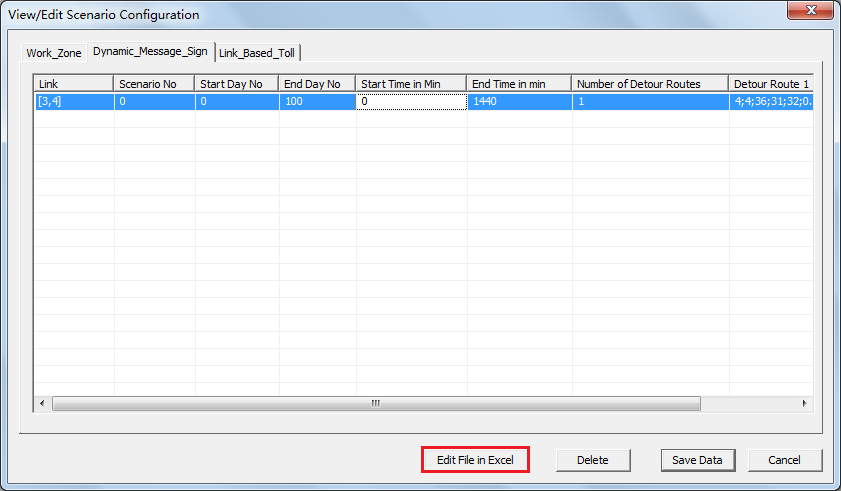


* Step 3-4: Gain the information of selected path in the opening file
* The necessary information data of selected path such as node sequence is noted by red block, as shown in the figure below
* Click the cell (noted by green block) and gain detail information of path node sequence(noted by red block above the figure)
* Count the number of nodes

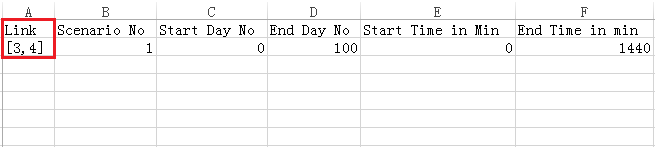


#### Step 4: Prepare Specific Input Files for VMS Scenario

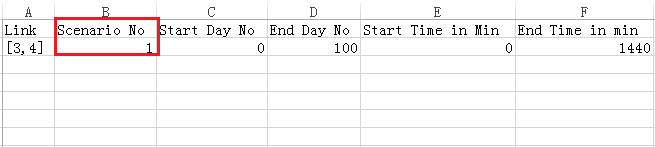
* Step4-1: Set  file
* Click button “Edit File in Excel” and Open file



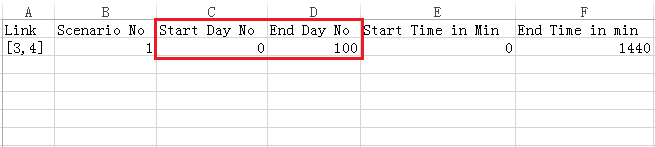
* Check origin node and destination node of VMS links



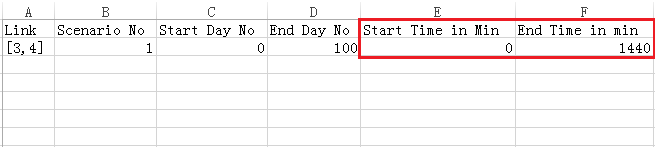
* Specify the sequence number of scenario (according to the value of “Scenario\_No” in file)



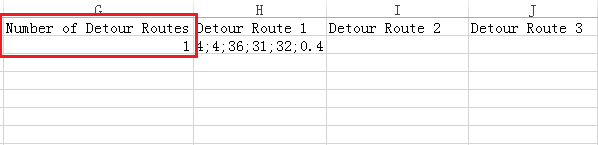
* Specify the iteration of VMS scenario
* In this example, the number of simulation iteration is set to “51” in file (more details in section **4.1. Basic Traffic Assignment Scenario**)
* The start day and end day of simulation iteration should be set to a large enough number to make sure VMS scenario is implemented under such conditions that dynamic traffic assignment and agents’ path choice in the whole network is stable (e.g., “Start Day”=50”and “End Day”=51 in this case)



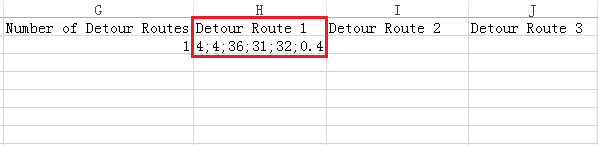
* Set the star time and end time of VMS scenario in specified iteration
* It should be noted that, VMS scenario should start after demand loading for a period of time.( According to  file)



* Specify the number of detour routes



* Set the number of nodes, node sequence, selective ratio of detour route
* It should be noted that the correct format of this cell is: [“the number of node in detour route”; “origin node”; “the second node”;….; “the destination node”; “selective ratio of detour route”], e.g., “4;4;36;31;32;0.4” in this case as shown in the figure below.

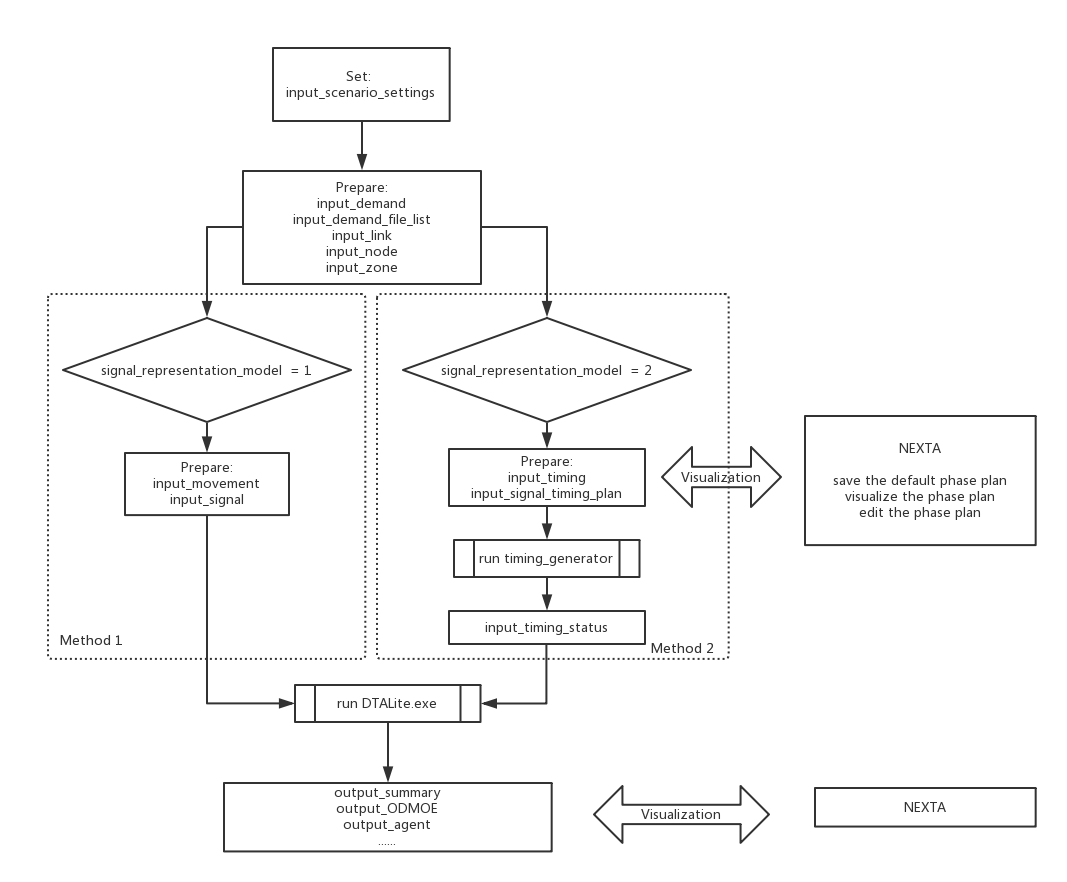


#### Step 5: Run Simulation and Perform Analysis

* Step 5-1: Double-click  after checking all input files, run simulation and perform analysis

## Signal Scenario (Link Based)

There are two models to deal with the signal timing in DTALite which are listed in the following flow charts. Method 1 is **link based signal representation model** and model 2 is **phase based signal representation model**.



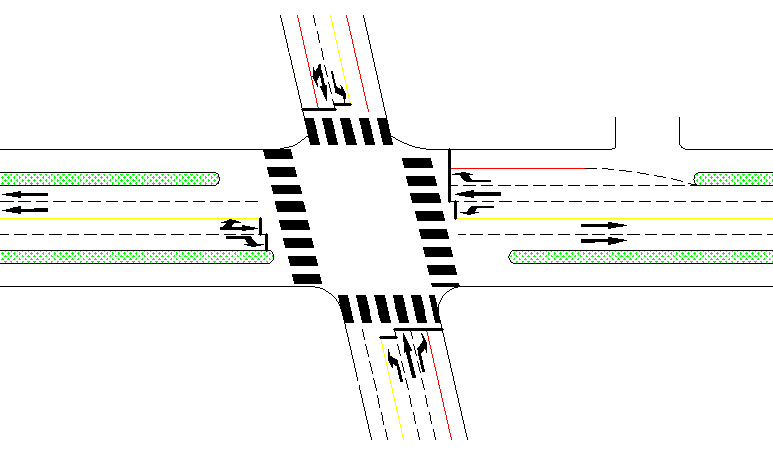
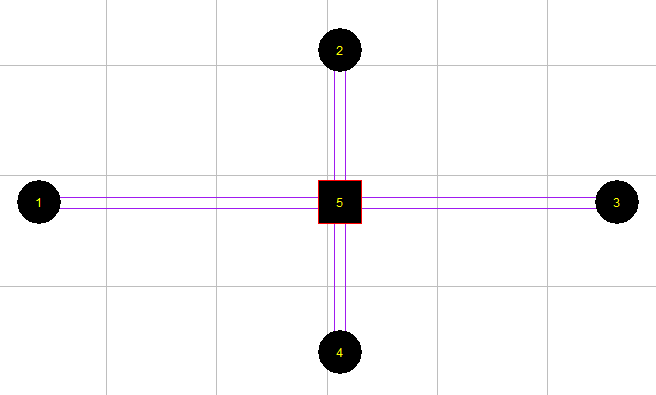
This section we introduce how to use method 1: link based signal representation model in the following steps.

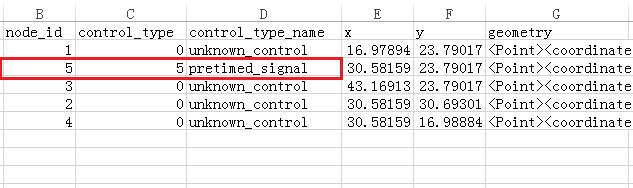
#### Step 1: Build a Basic Network and Prepare Necessary Input Files for Basic Scenario

See more details in section **4.1. Basic Traffic Assignment Scenario**.

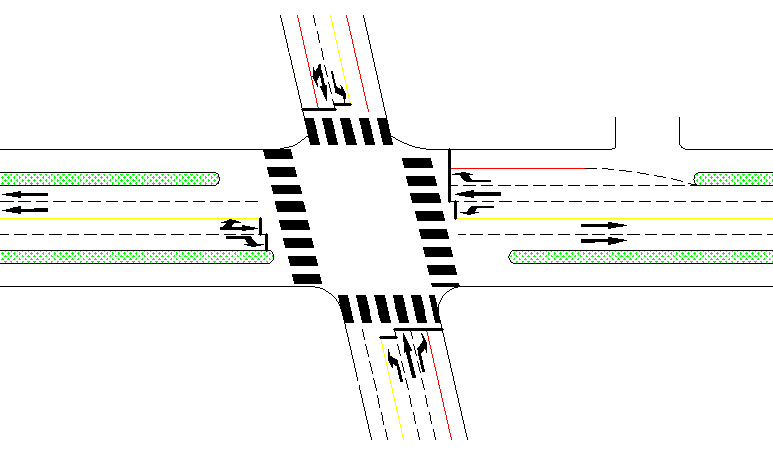
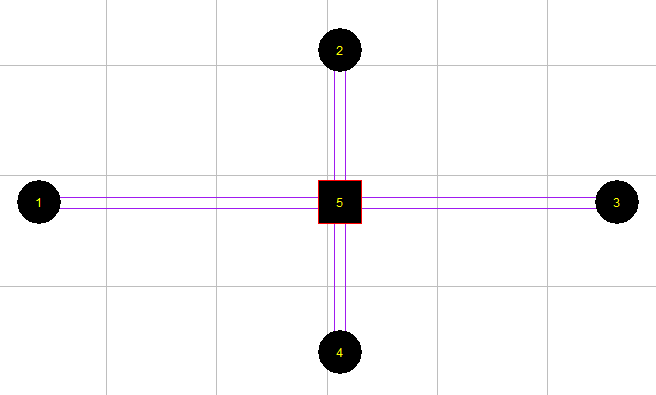
Additionally, It should be noted that:

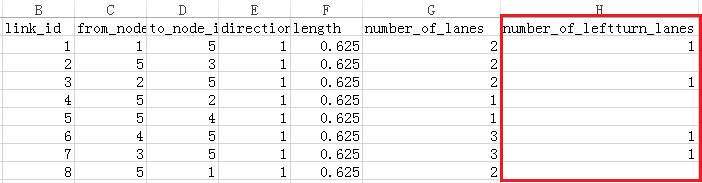
* In file
* Set the control type of signal nodes to “5>pretimed signal” (e.g., the control type of node 5 is set to “pretimed signal” in this case)

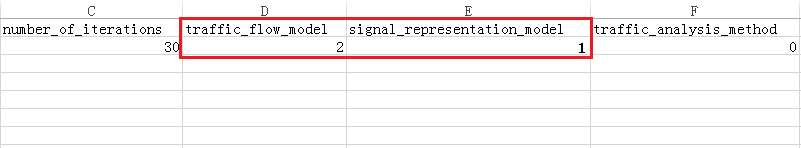


* In file
* Specify the number of left-turn lanes (including separated left-turn lanes and shared left-turn lanes) according to geometry construction of intersection (e.g., in this case, the number of left-turn lane of link 1 is set to “1”)

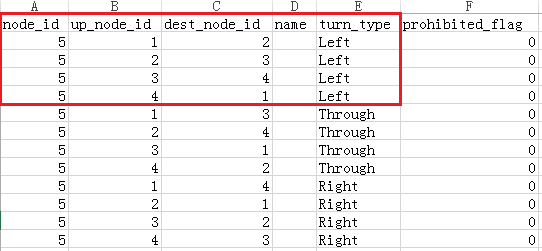


* In file
* Set traffic flow model to “2> Spatial Queue Model”
* Set signal representation model to “1> Cycle Length+ Link-based Effective Green Time”

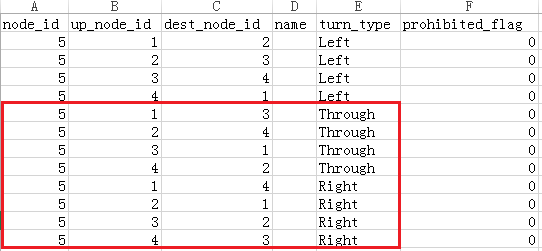


#### Step 2: Prepare Specific Input File for Signal Scenario

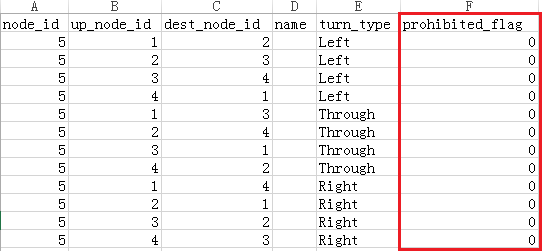
* Step 2-1: Set  file
* Specify left-turn type movement at signal nodes which is required



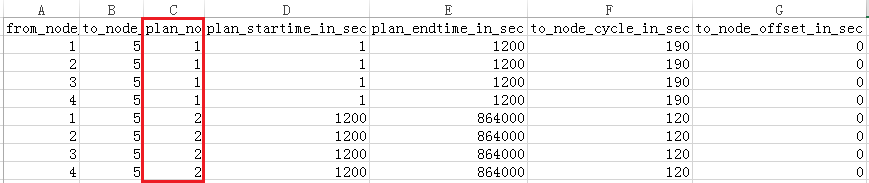
* Specify through type movement and right-turn movement at signal nodes which is desired



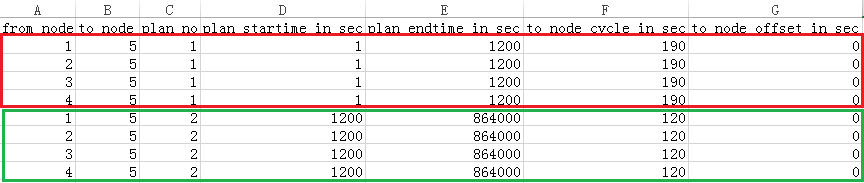
* Set prohibited flag to “1” when movement is prohibited, or else set prohibited flag to “0”



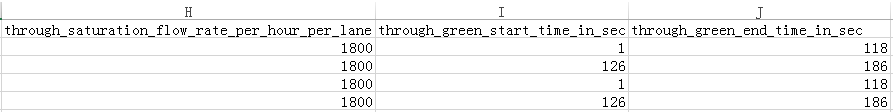
* Step 2-2: Set  file
* Set the number of phase plan



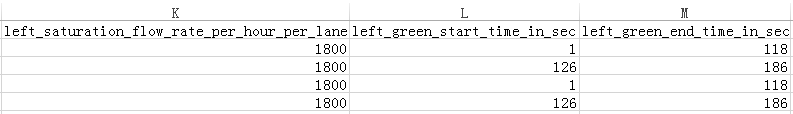
* Set starting time(in seconds), ending time(in seconds), cycle length(in seconds) and offset(in seconds) of corresponding phase plan, across all link based records associated with this plan at this signal node
* “Offset” is used to denote the difference of green start time between adjacent intersections in signal cooperative control, one can use any default values in one-point controlled signal



* Specify through saturation flow(per hour per lane), start time(in seconds) and end time(in seconds) for through green phase (e.g., phase plan 1 is shown in the figure below)



* Specify left-turn saturation flow(per hour per lane), start time(in seconds) and end time(in seconds) for left-turn green phase (e.g., phase plan 1 is shown in the figure below)



#### Step 3: Run Simulation and Perform Analysis

* Step 3-1: Double-click  after checking all input files, run simulation and perform analysis

## Signal Scenario (Phase Based)

#### Section 1: Key files required for signal scenario settings

The key input setting files and output files for signal scenario are shown in Table 1.

**Table 1. Key files for signal scenario settings**

|  |  |  |
| --- | --- | --- |
| **File Group** | **Files list** | **Remark** |
| Input files | 1. input\_demand.csv | Demand content for specifying distribution (observed flows of intersections) |
| 2. input\_demand\_file\_list.csv | Demand files specification, loading multiplier, specify departure time distribution |
| 3. input\_zone.csv | TAZ definition |
| 4. input\_link.csv | Properties of all links in network |
| 5. input\_node.csv | Properties of all nodes in network |
| **6.input\_timing.csv\*** | Definition of the phase in each timing plan |
| **7.input\_signal\_timing\_plan.csv\*** | Definition of the planning of intersections in one day |
| **8.input\_timing\_status.csv\*** | The second to second phase plan of every intersection, which will be automatically generated by the **timing\_generator.exe** |
| 9. input\_scenario\_settings.csv | Specify signal representation model and assignment settings. **signal\_representation\_model** should be set in **input\_scenario\_settings.** |
| …… |  |
| Output files | 1. output\_agent.csv | Statistics of agents such as path node/time sequence, OD, departure/arrival time |
| 2. output\_summary.csv | General statistics of simulation results for each iteration |
| ...... |  |

**\*6, 7, 8 input\_timing.csv, input\_signal\_timing\_plan.csv, input\_timing\_status.csv** are the new input files.

There are altogether 3 kinds of signal timing controlling methods inside the DTALite, namely:

* signal\_model\_continuous\_flow = 0, the first signal model (signal\_model\_continuous\_flow) doesn’t consider signal;
* signal\_model\_link\_effective\_green\_time = 1, the link based signal representation model, need input\_signal.csv and input\_movement.csv;
* signal\_model\_timing\_status = 2, the Phase Based Signal Representation Model takes the input\_timing.csv, input\_signal\_timing\_plan.csv and the input\_timing\_status.csv as the input files;

Table 2 below gives an example of the parameter settings inside input\_scenario\_settings.csv.

**Table 2. Parameter settings in file input\_scenario\_settings.csv**

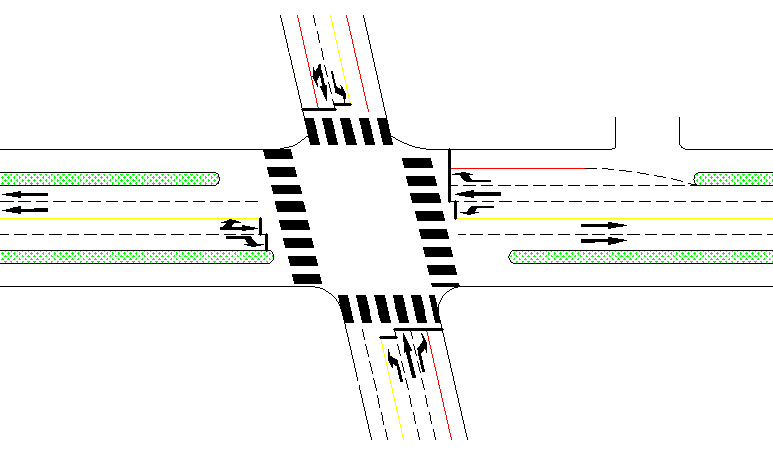
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **scenario\_no** | **scenario\_name** | **number\_of\_iterations** | **traffic\_flow\_model** | **signal\_representation\_model** | **traffic\_analysis\_method** |
| 1 | test1 | 10 | 2 | 2 | 0 |

**Remark:**

* the signal\_representation\_model of ”2” is required to enable signal scenario assignment.
* “number\_of\_iterations” defines the total numbers of iterations for scenarios corresponding to “scenario\_no”.
* the traffic\_flow\_model of ”2” is dedicated to spatial queue model.

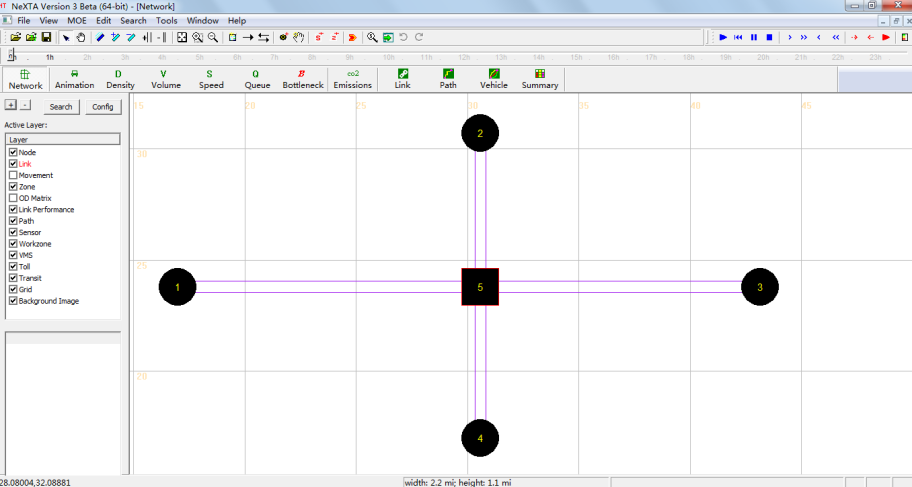
#### Section 2: Input data preparation

A fixed-time controlled signal is chosen as a simple case in this guide. The detailed information is shown in figure 2 and table2. (dataset stored in 0-Signal\_test\_dataset\_5\_node\_network)



**Figure 2. Geometry construction of sample intersection**

In this case, the traffic network including 5 nodes and 8 links are built in NEXTA, as shown in Figure 3.



**Figure 3. Traffic network displayed in NeXTA**

As described above, the required basic input data files for signal scenario include the demand data (input\_demand.csv and input\_demand\_file\_list.csv) and the traffic network data (input\_node.csv, input\_link.csv, input\_zone.csv). The detailed format of input files based on sample data are listed in Table 3-7.

**Table 3. Sample data in file input\_demand.csv**

|  |  |  |
| --- | --- | --- |
| **from\_zone\_id** | **to\_zone\_id** | **number\_of\_trips\_demand\_type1** |
| 1 | 2 | 100 |
| 1 | 3 | 100 |
| 1 | 4 | 100 |
| 2 | 3 | 100 |
| 2 | 4 | 100 |
| 2 | 1 | 100 |
| 3 | 4 | 100 |
| 3 | 1 | 100 |
| 3 | 2 | 100 |
| 4 | 1 | 100 |
| 4 | 2 | 100 |
| 4 | 3 | 100 |

**Table 4. Sample data in file input\_demand\_file\_list.csv**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **scenario\_no** | **file\_sequence\_no** | **file\_name** | **Format\_type** | **Loading\_multiplier** | **start\_time\_in\_min** | **end\_time\_in\_min** | **number\_of\_demand\_types** |
| 1 | 1 | input\_demand.csv | column | **1** | **0** | **60** | 1 |

**Table 5. Sample data in file input\_zone.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **zone\_id** | **production** | **attraction** | **geometry** |
| 1 | 0 | 0 | <Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon> |
| 2 | 0 | 0 | <Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon> |
| 3 | 0 | 0 | <Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon> |
| 4 | 0 | 0 | <Polygon><outerBoundaryIs><LinearRing><coordinates></coordinates></LinearRing></outerBoundaryIs></Polygon> |

**Table 6. Sample data in file input\_link.csv**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **from\_node\_id** | **to\_node\_id** | **length** | **number\_of\_lanes** | **number\_of\_leftturn\_lanes** | **speed\_limit** | **jam\_density** |
| 1 | 5 | 0.625 | 2 | **1** | 25 | 200 |
| 5 | 3 | 0.625 | 2 |  | 25 | 200 |
| 2 | 5 | 0.625 | 2 | **1** | 25 | 200 |
| 5 | 2 | 0.625 | 1 |  | 25 | 200 |
| 5 | 4 | 0.625 | 1 |  | 25 | 200 |
| 4 | 5 | 0.625 | 3 | **1** | 25 | 200 |
| 3 | 5 | 0.625 | 3 | **1** | 25 | 200 |
| 5 | 1 | 0.625 | 2 |  | 25 | 200 |

**Remark:**

* Number of lanes: the number of lanes on specific links.
* The value of “Number of left-turn lanes” should be specified when left-turn lanes exist (including separated left-turn lanes and shared left-turn lanes)

**Table 7. Sample data in file input\_node.csv**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **node\_id** | **control\_type** | **control\_type\_name** | **x** | **y** | **geometry** |
| 1 | 0 | unknown\_control | 16.97894 | 23.79017 | <Point><coordinates>16.978937,23.790170</coordinates></Point> |
| 5 | **5** | pretimed\_signal | 30.58159 | 23.79017 | <Point><coordinates>30.581594,23.790170</coordinates></Point> |
| 3 | 0 | unknown\_control | 43.16913 | 23.79017 | <Point><coordinates>43.169128,23.790170</coordinates></Point> |
| 2 | 0 | unknown\_control | 30.58159 | 30.69301 | <Point><coordinates>30.581594,30.693011</coordinates></Point> |
| 4 | 0 | unknown\_control | 30.58159 | 16.98884 | <Point><coordinates>30.581594,16.988842</coordinates></Point> |

**Remark:**

* The control type of intersection nodes should be specified as “pretimed signal” (as shown node 5 in the table above). 0 here means unknown control.
* For the detailed node type information, please check input\_node\_control\_type.csv.

#### Section 3: Parameter settings in key simulation configuration files

First, as mentioned in table 2, set **signal\_representation\_model = 2** in input\_scenario\_settings.csv.

Then, prepare **the input\_signal\_timing\_plan.csv** and **input\_timing.csv** tables.

The input\_signal\_timing\_plan.csv file looks like this:

**Table 8. Sample data in file input\_signal\_timing\_plan.csv**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **[TOD]** | **int\_id** | **timing\_plan\_no** | **start\_time\_in\_sec** | **end\_time\_in\_sec** | **starting\_phase** | **off\_set** |
|  | 5 | 1 | 0 | 86400 | 1 | 11 |

**Remark:**

* **int\_id**: intersection node id;
* **timing\_plan\_no**: the timing plan of this intersection, for example, we could have different phase plans in one intersection on peak hour and off-peak hour (could define maximum 20 different timing plans in one day). It should have the same number of timing plans for an intersection with the file of input\_timing (will be introduced next);
* **start\_time\_in\_sec:** the start time of the timing plan. (unit is seconds);
* **end\_time\_in\_sec:** the end time of the timing plan. (unit is seconds);
* **starting\_phase:** the starting phase of this timing plan;
* **off\_set:** the time relationship between coordinated.

The **input\_timing.csv** looks like this:

**Table 9. Sample data in file input\_timing.csv**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **[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; |

**Remark:**

* **int\_id**: intersection node id;
* **timing\_plan\_no**: the timing plan of this intersection, for example, we could have different phase plans in an intersection on peak hour and off-peak hour (could define maximum 20 different timing plans in one day);
* **phase\_id**: in one cycle, there are several phases, usually we use NEMA phase plans with phase\_id from 1 to 8. (could define maximum 20 different phases in one timing plan);
* **next\_phase:** the phase\_id of the next phase;
* **green\_duration:** the green duration of this phase (here doesn’t consider the all read and yellow time);
* **movement\_str:** this is the most crucial element in this table. Here, we have the start\_node\_id and end\_node\_id of this phase;
* **movement\_dir\_str:** is the direction string of the movements, which we could get directly from Synchro data.

Using the input\_timing.csv file and input\_signal\_timing\_plan.csv file, we could generate the input\_timing\_status.csv file with **timing\_generator.exe** :

The input\_timing\_status.csv file looks like this:

**Table 10. Sample data in file input\_timing\_status.csv**

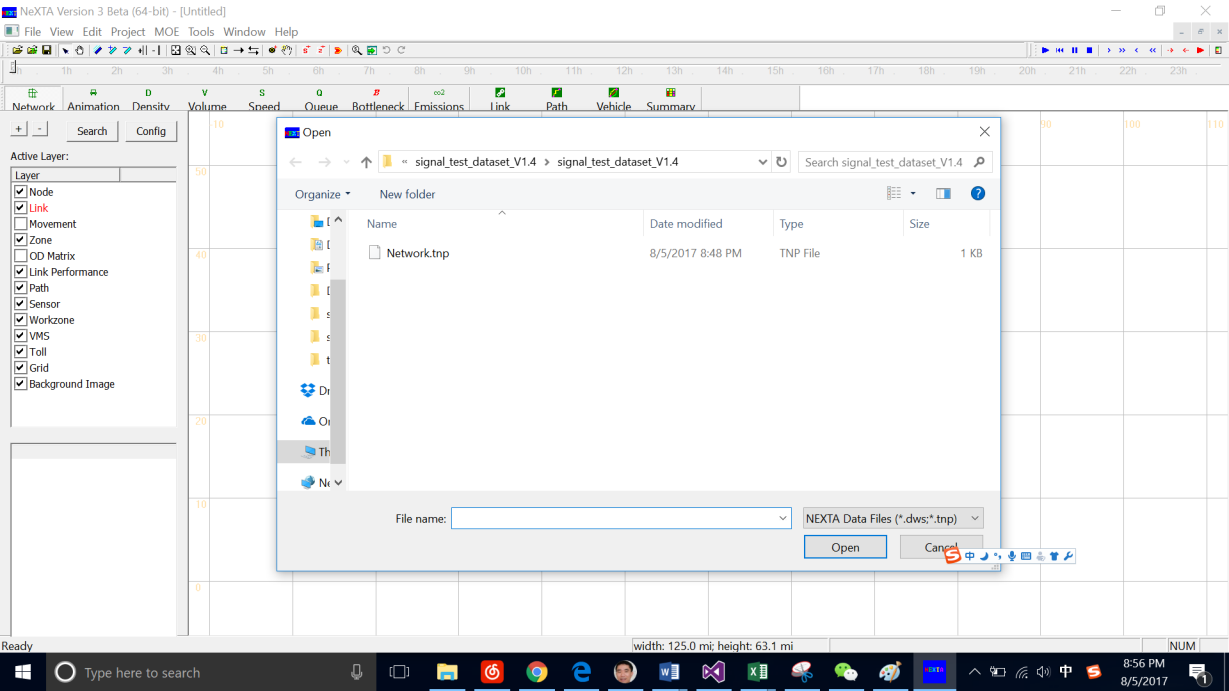
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **movement\_str** | **start\_time\_in\_sec** | **end\_time\_in\_sec** | **signal\_status** | **from\_node\_id** | **to\_node\_id** | **turn\_type** |
| 1\_3\_T | 1 | 120 | 1 | 1 | 5 | T |
| 1\_2\_L | 1 | 120 | 1 | 1 | 5 | L |
| 3\_1\_T | 1 | 120 | 1 | 3 | 5 | T |
| 3\_4\_L | 1 | 120 | 1 | 3 | 5 | L |
| 2\_4\_T | 121 | 190 | 1 | 2 | 5 | T |
| 2\_3\_L | 121 | 190 | 1 | 2 | 5 | L |
| 4\_2\_T | 121 | 190 | 1 | 4 | 5 | T |
| 4\_1\_L | 121 | 190 | 1 | 4 | 5 | L |
| … | … | … | … | … | … | … |

* **start\_time\_in\_sec:** the start time of each small phase;
* **end\_time\_in\_sec:** the end time of each small phase;
* **signal\_status:** 1: green; 2: red;
* **from\_node\_id:** from\_node\_id of this phase;
* **to\_node\_id**: it represents the intersection id of this phase, (different from the movement\_str in input\_timing). We generate it in this way because, there are only real links between normal nodes and the intersection nodes. For example, when the movement string movement\_str is 1\_3\_T, there is no direct link from node 1 to node 3, it is actually 1->5->3. Therefore, the to\_node\_id is 5 and the turn type is T (Through).
* **turn\_type:** the phase direction, **T** means through and **L** means left turn.
* After preparing all these files, we copy them to the working folder with DTALite.exe and other input files.

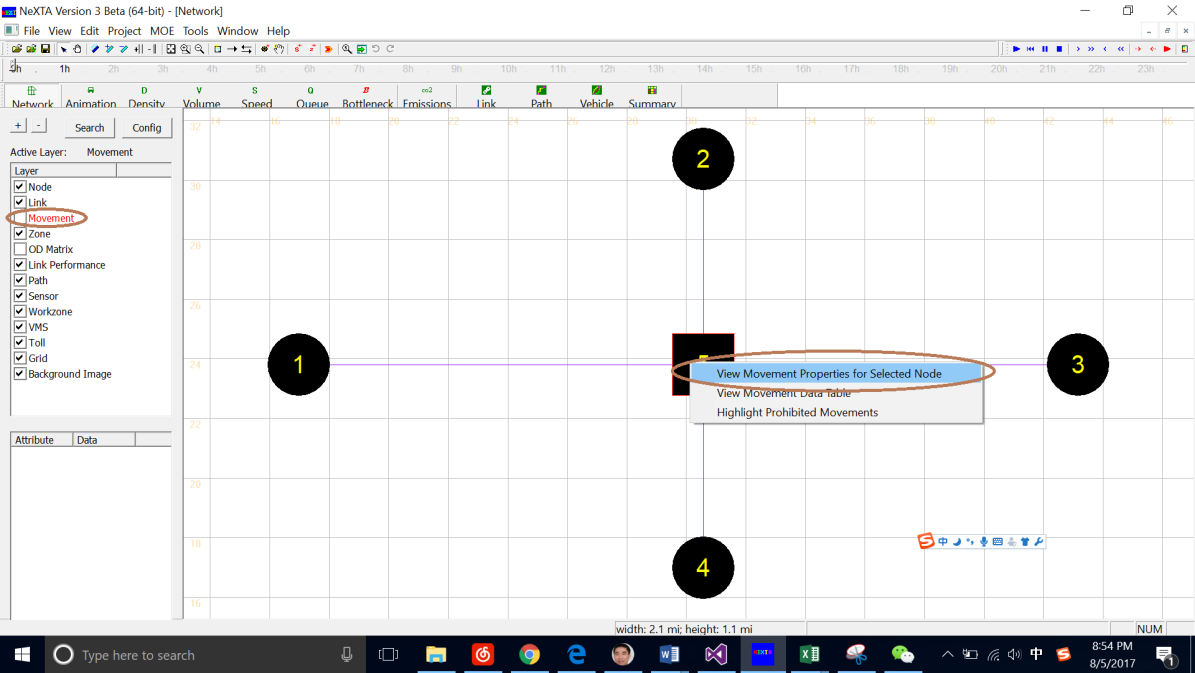
#### Section 4: NEXTA GUI instruction

The NEXTA GUI now could visualize the phase plans of each intersection.

* Load the example project:



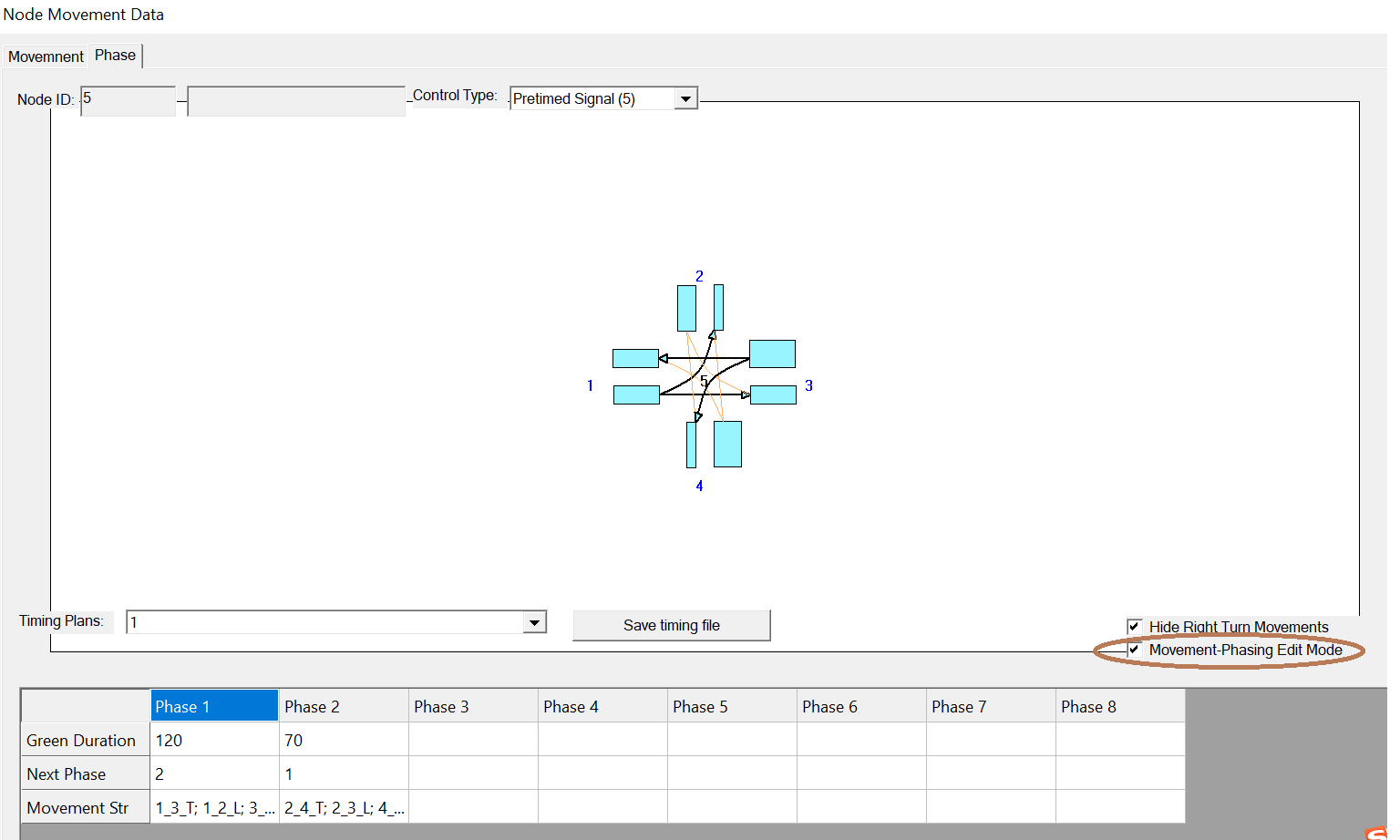
* Choose the **Movement** layer -> right click the intersection node -> View Movement Properties for Selected Node.



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

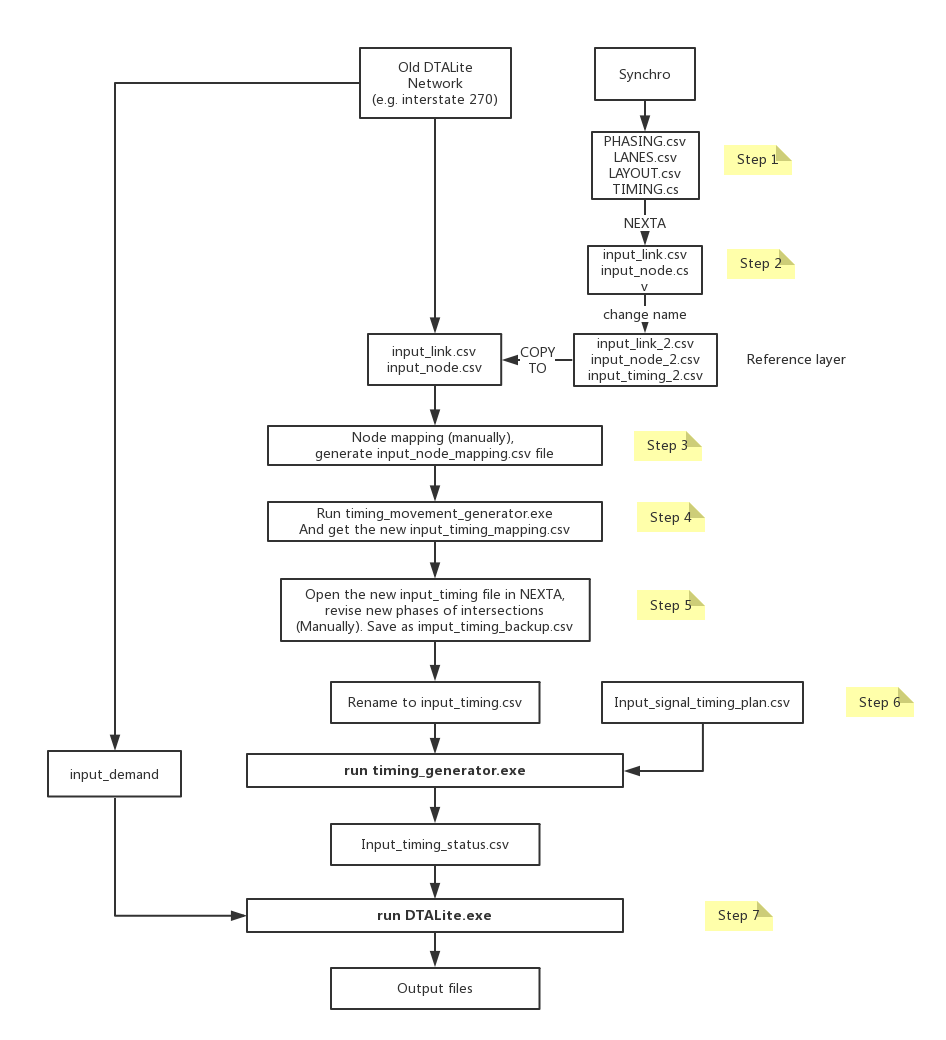
If we want to add a phase in current phase, we need to check the box before **Movement-Phasing Edit Mode**. Then double click the phase the you want to add, then the corresponding movement\_str will be added to the table below. If we want to delete the phases added, we could double click it again. Thus, the Click save **timing file button**, the **new input\_timing** file will be in the folder.

For example:



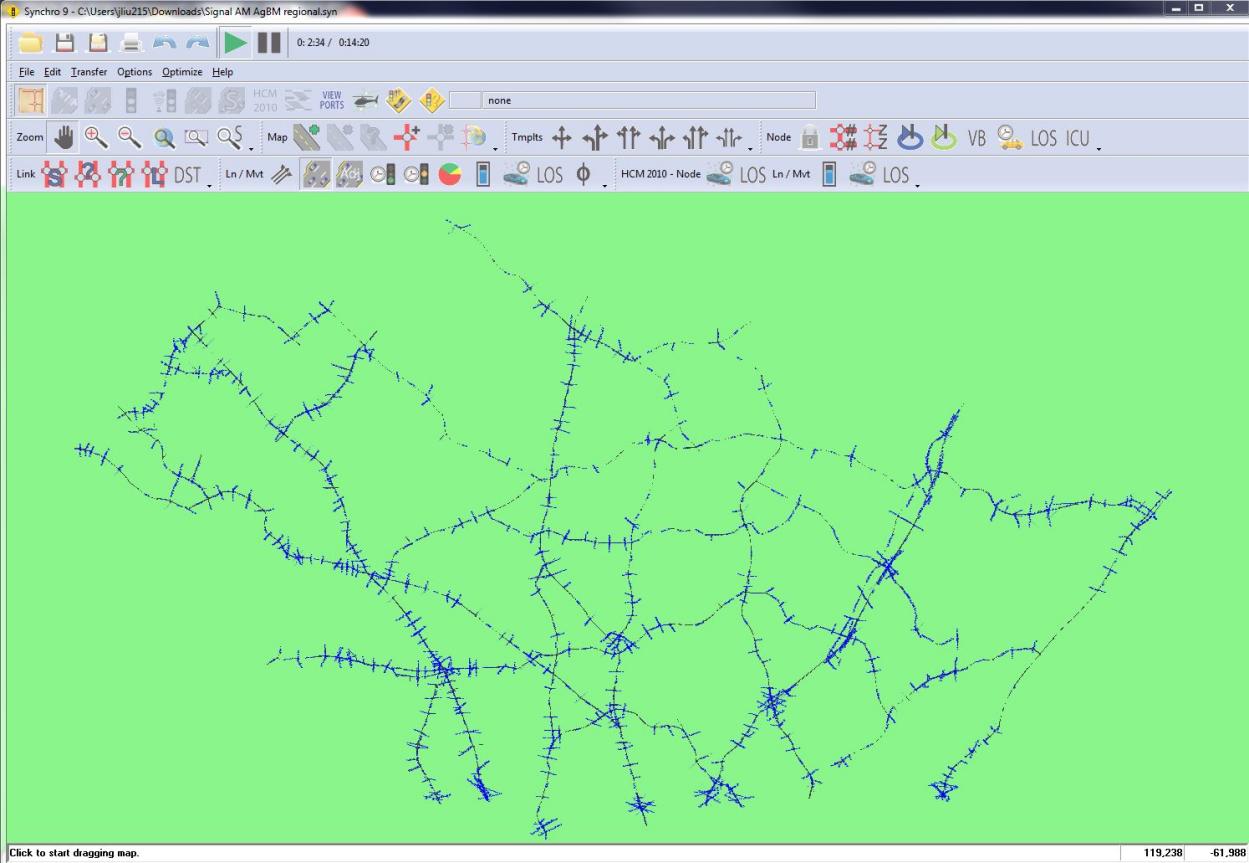
#### Section 5: Step by step instruction of how to import data from Synchro to DTALite

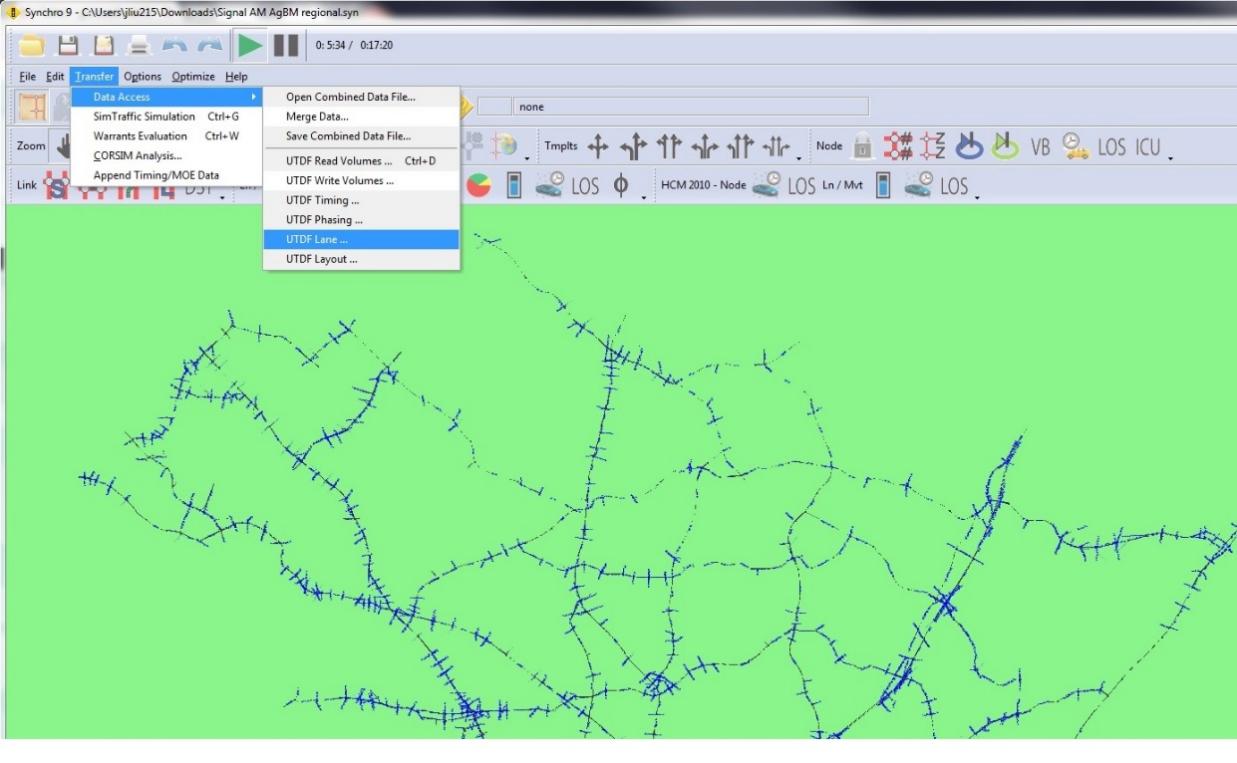
The flow chart of this seven-step procedure of this function:

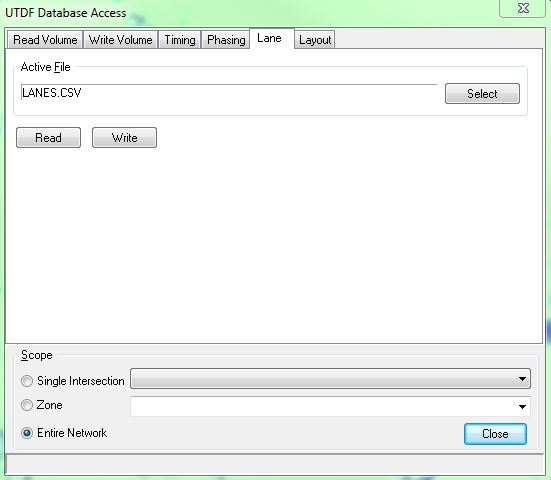


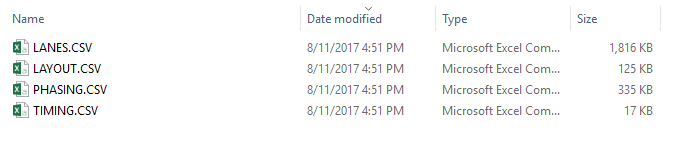
* **Step 1: Export csv files from Synchro**

Prepare the Synchro dataset, export to LANES.cvs, LAYOUT.csv, PHASING.csv and TIMING.CSV (UTDF format). The Synchro data is stored in 1-Signal Timing Data\_from Synchro.

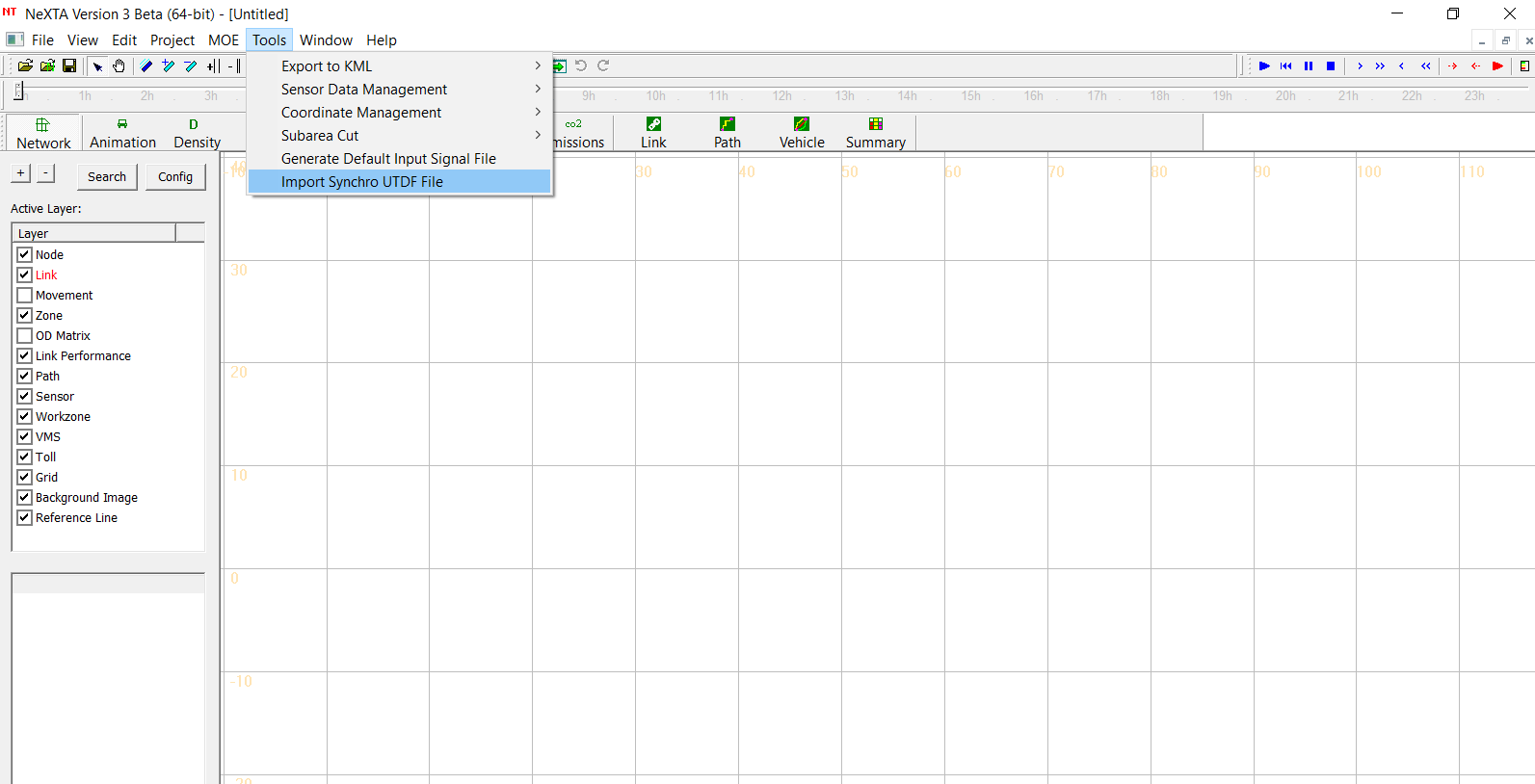








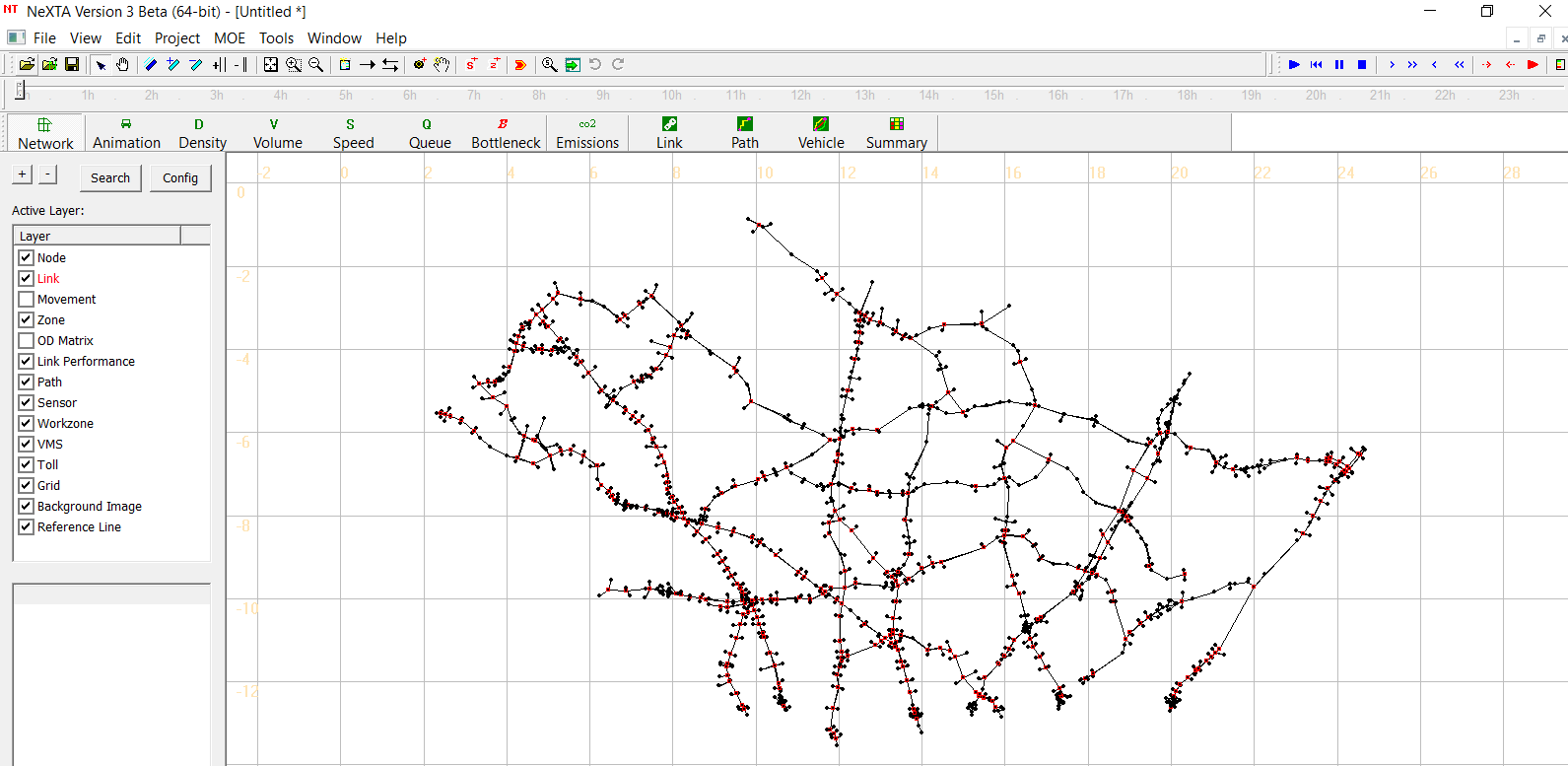
* **Step 2: Use NEXTA to import the Synchro UTDF format**
* Open NEXTA, click tools -> Import Synchro UTDF File:



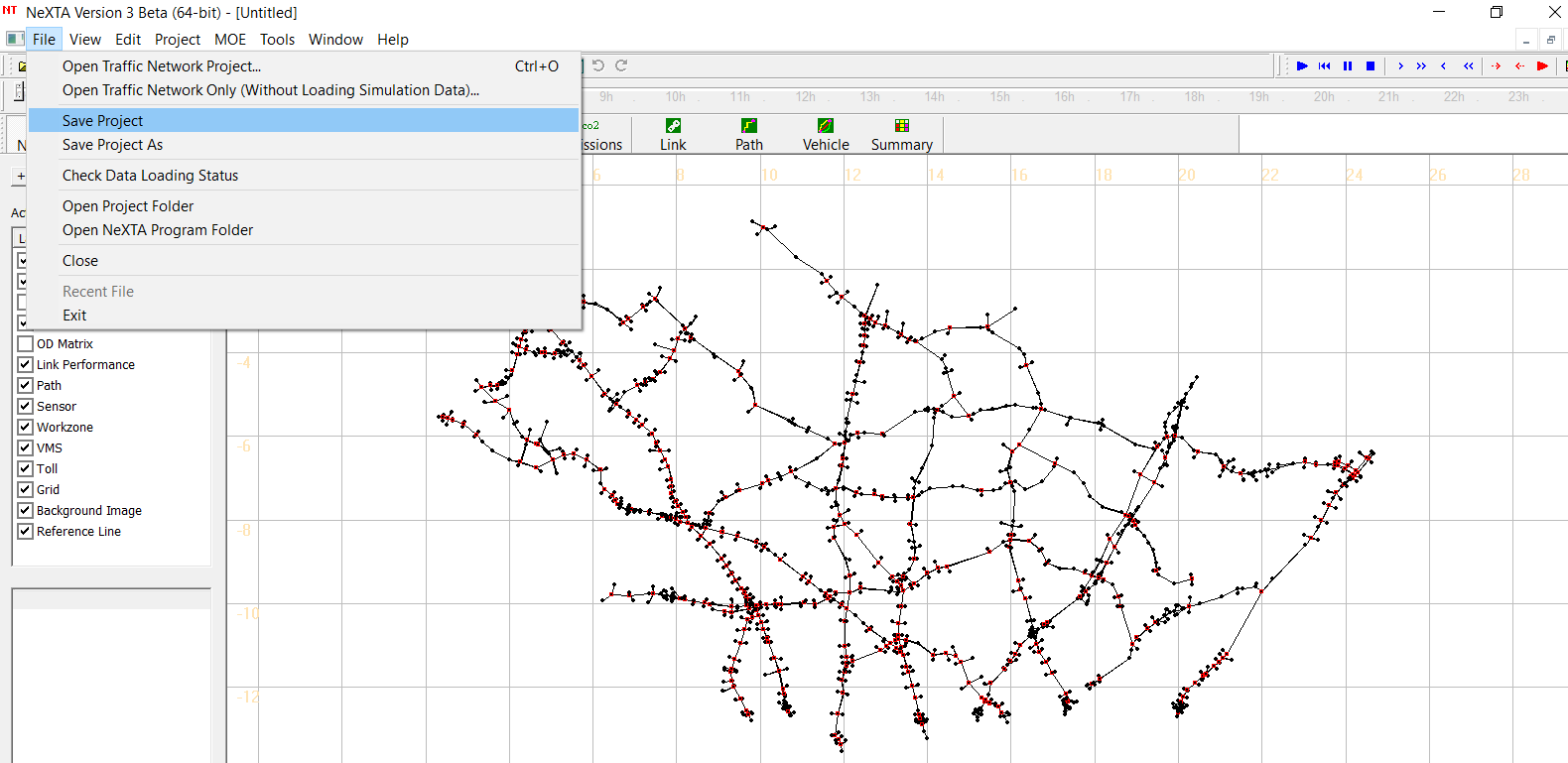
* Open LAYOUT.csv file that saved above:

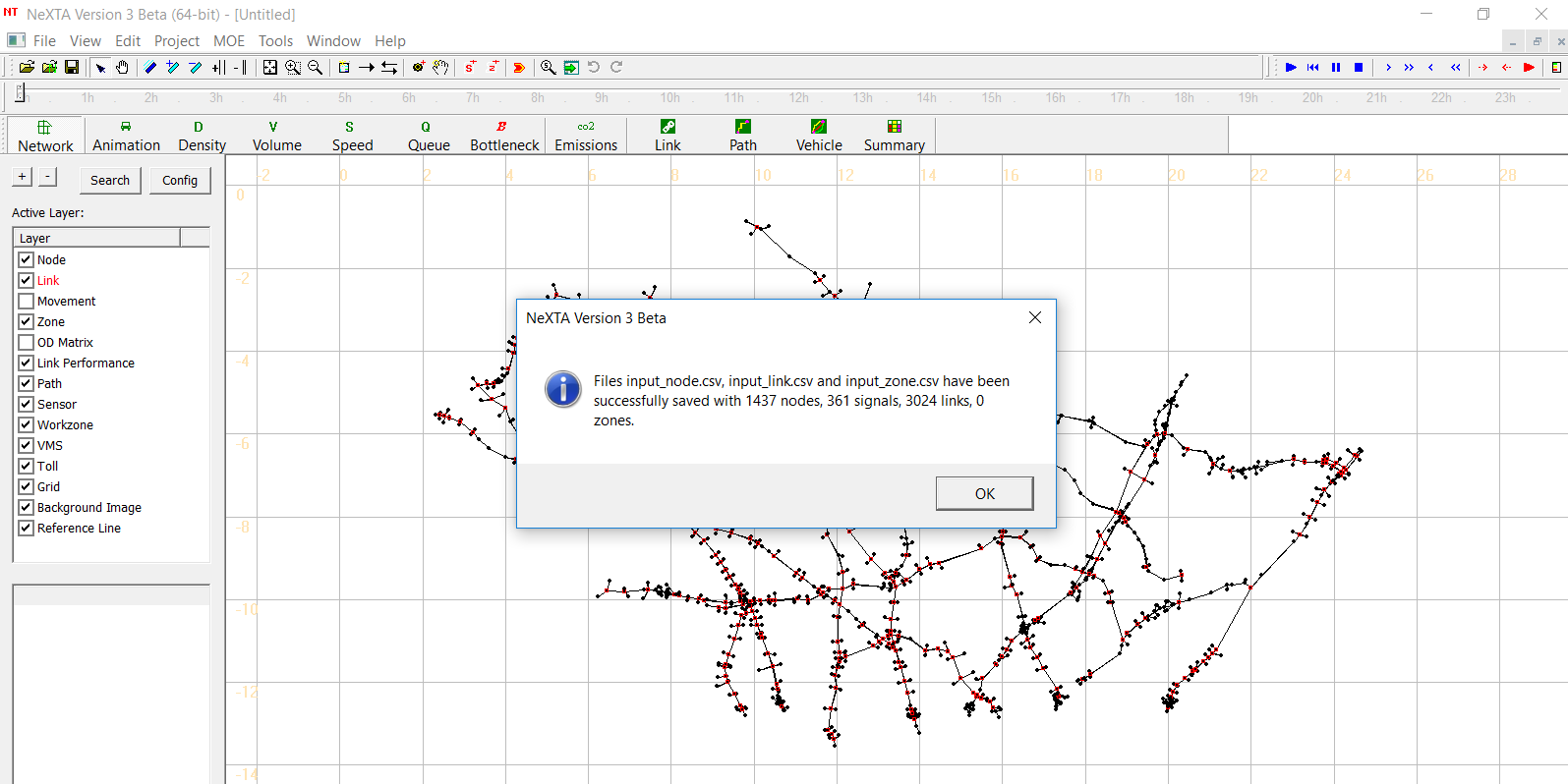


* The synchro data will be imported to NEXTA.



* Save the project in the same folder.





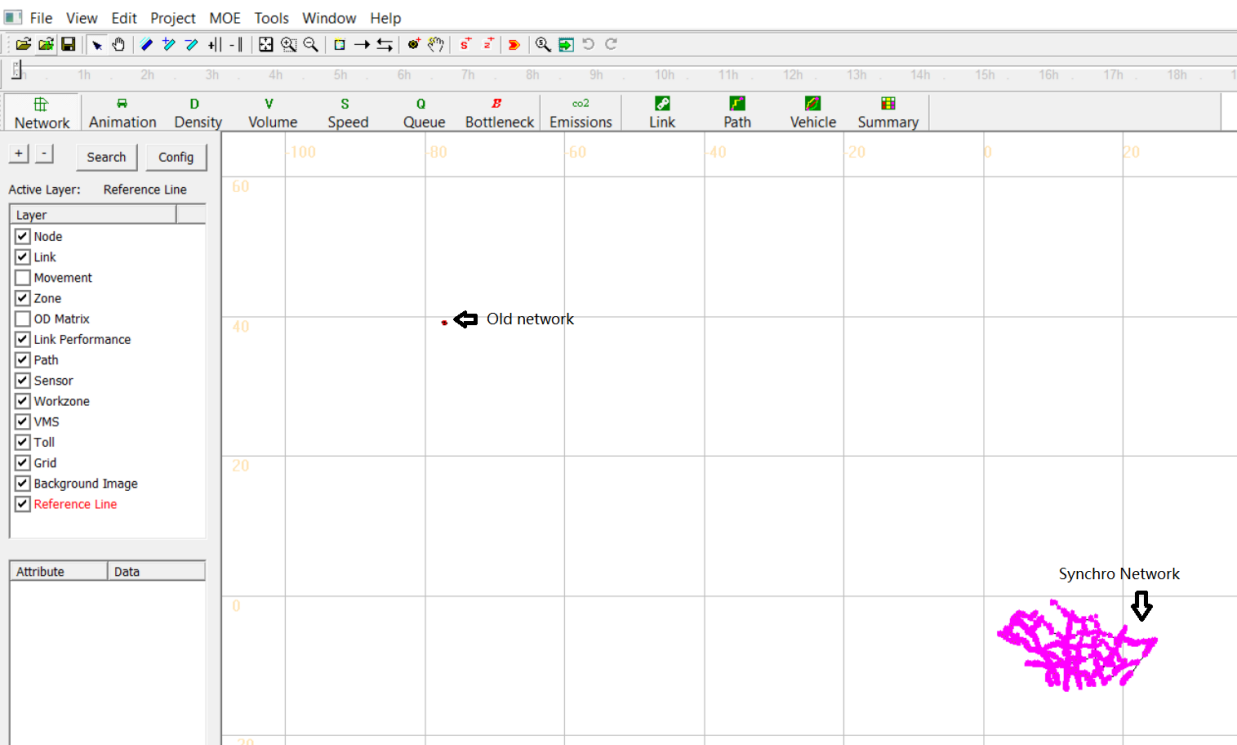
The network and the signal data will be automatically saved into the folder. You could find the data in 2-Interstate\_270\_synchro\_export.

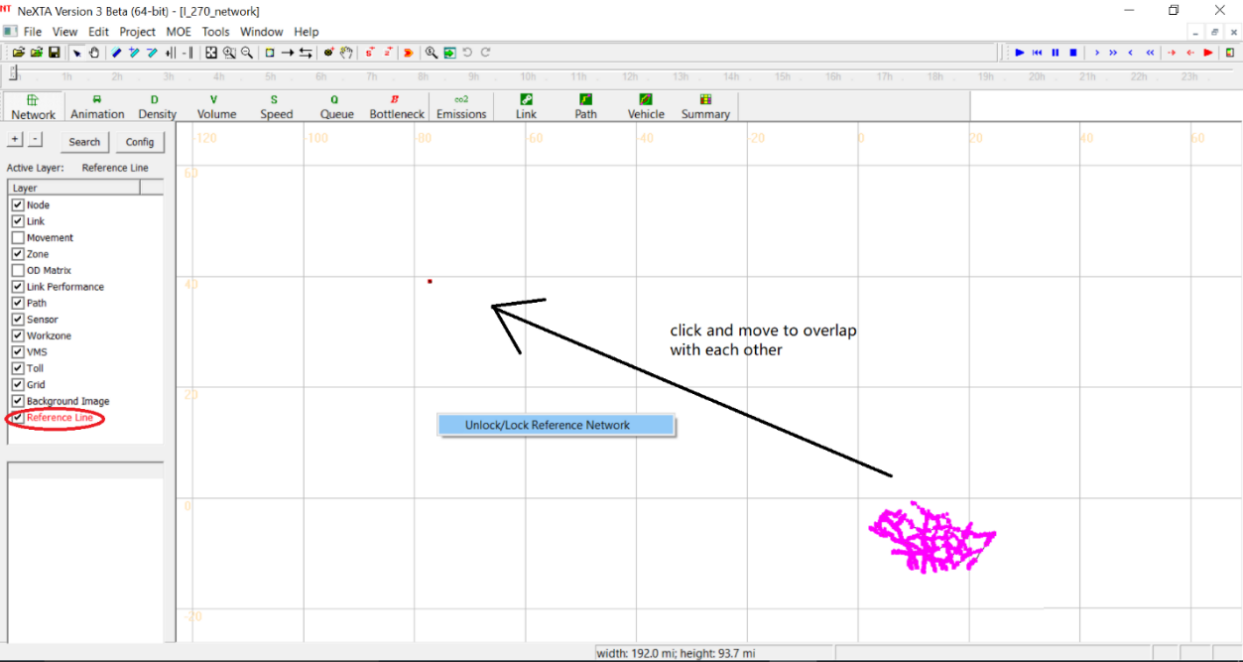


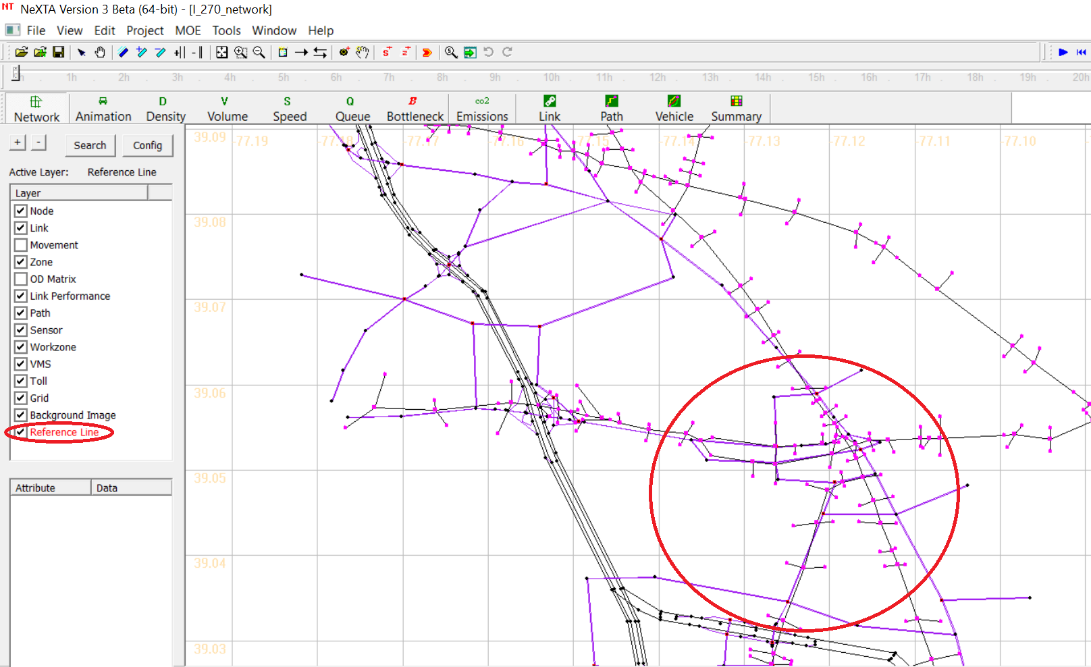
* **Step 3: Mapping two networks**

In this step, we have two networks (a) DTALite main network (b) the imported synchro network in DTALite format. Use DTALite to shift the network (b) toward main network (a) so we can build the mapping between two networks (originally in different coordinator). Nexta has a tool to display all mapped intersections to verify the results.

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. Then, open the new project in NEXTA. Data could be found in 3-Interstate\_270\_Network – mapping.







But now the **node and Link ID** of the two networks are not matched with each other, we need to map them with our old DTALite Network.

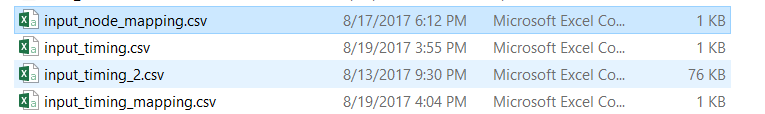
Based on this, we could have the **input\_node\_mapping**.csv table generated by ourselves manually.

|  |  |
| --- | --- |
| **node\_id\_1** | **node\_id\_2** |
| 3346 | 251 |
| 3335 | 1208 |
| 3338 | 1505 |
| … | … |

**Remark:**

* **Node\_id\_1** is the node id from the old DTALite network, **node\_id\_2** is the mapping node from synchro exported network.
* **Step 4: Generate the new input\_timing\_mapping.csv**

Run **timing\_movement\_generator.exe** to generate the new input\_timing\_mapping.csv of phasing plan combination for both networks.



We create the **input\_node\_mapping**.csv file in step 3 and then copy the **input\_timing\_2**.csv from the Synchro to the new folder. Then we could run the simulation of **timing\_movement\_generater.exe** and get the result of input\_timing\_mapping.csv:



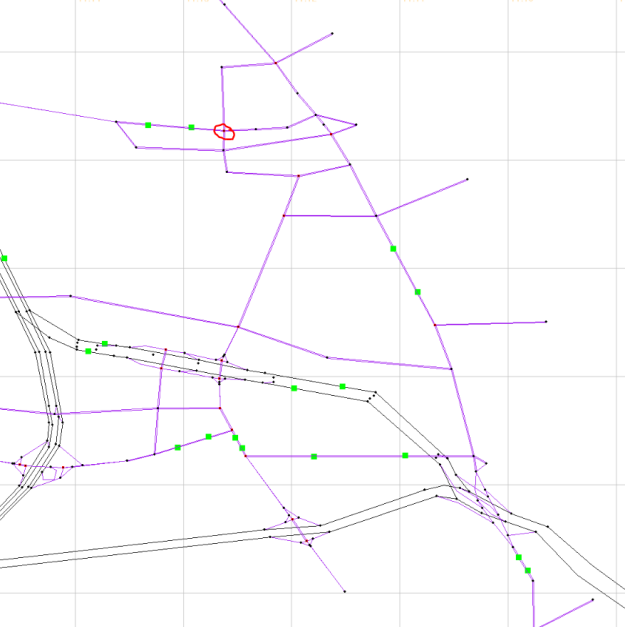
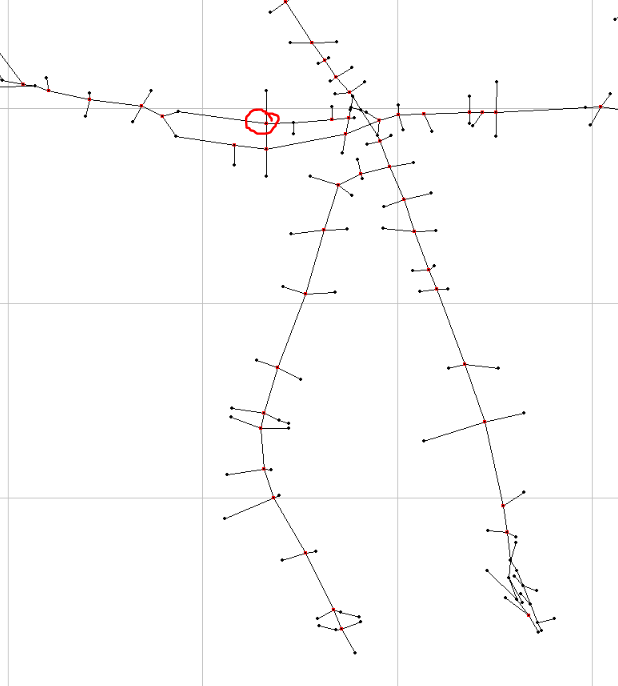
Copy this input\_timing\_mapping.csv to the old DTALite network folder and then replace the old input\_timing.csv file. Meanwhile, need to change the corresponding input\_signal\_timing\_plan.csv file to match the total number of intersections and phase plans.

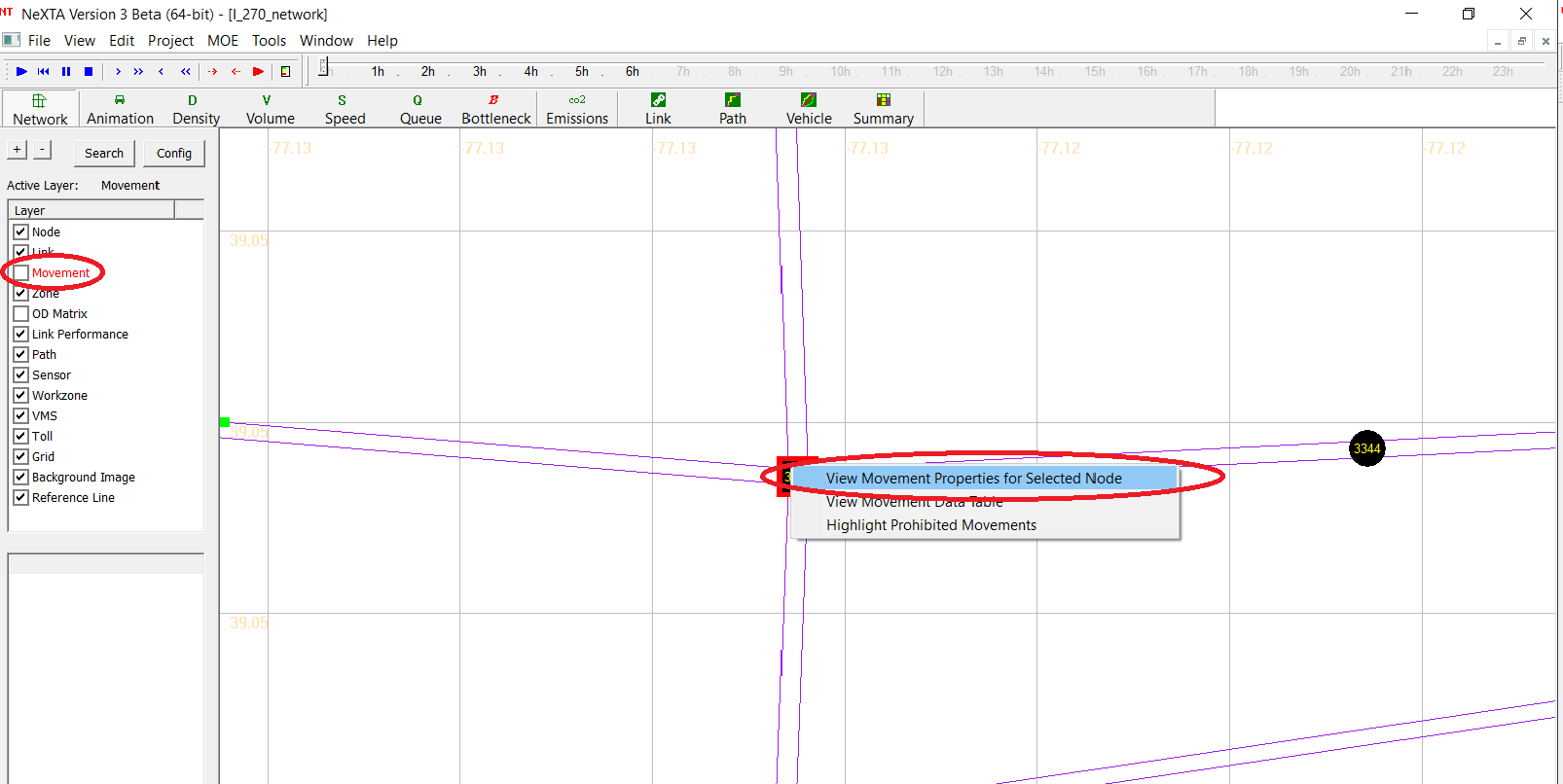
* **Step 5: Verify the input phasing results**

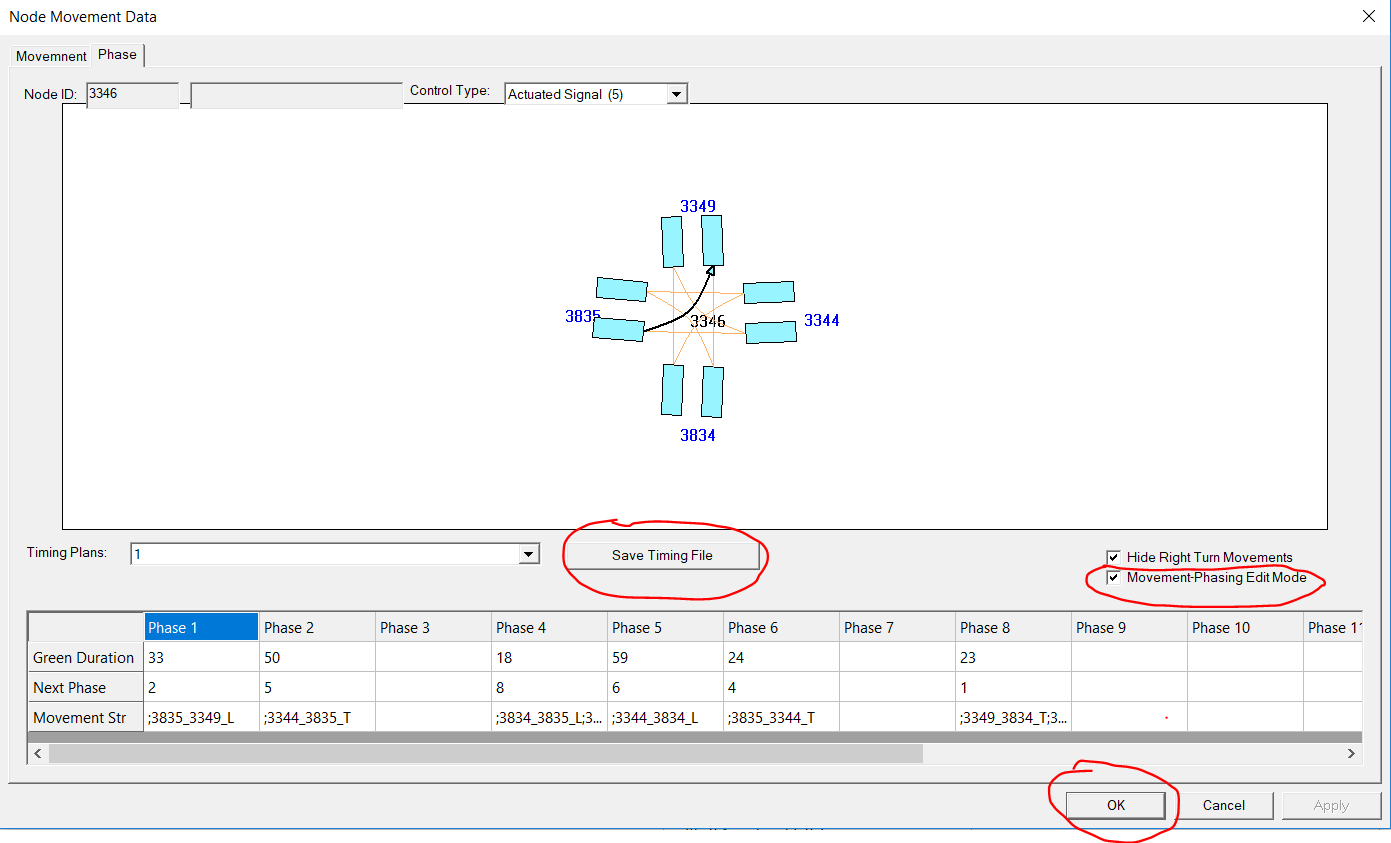
Verify the input phasing results and add required next phase number. In main DTA network, make sure the imported phasing data are correctly display in NEXTA.

If they are not well updated, we need to manually update the phase plans.

Then, re-open the network in NEXTA and you will see the two networks in NEXTA.





We manually check the accuracy of the phase data and make some adjustments if needed. After validation, it will be saved as input\_timing.csv

* **Step 6: Generate input\_timing\_status.csv file**

Then, click the timing\_generator.exe to generate the input\_timing\_status.csv file which will be used in DTALite.exe for simulation.

* **Step 7: Run DTALite.exe**

Run DTALite.exe to get the output results.

#### Section 6: Key aspects

* Make sure the signal\_representation\_model in input\_scenario\_settings are set to be the right number, for example, “2” for Phase Based Signal Representation Model which we mainly discussed in the documentation.
* Make sure the demand file is correctly set in input\_demand\_file\_list.csv.
* Before adding new phase plans in the network, change the node\_type in input\_node.csv to “5”, the actuated signal.
* The timing\_generator.exe could generate the input\_timing\_status.csv on both NEMA phase data and other phase data. However, make sure for the NEMA phase plans, all the next phase ids are set to be “-1”.
* Make sure that the input\_timing.csv and input\_signal\_timing\_status.csv have the same amount of signal intersections and same timing plans for each intersection.

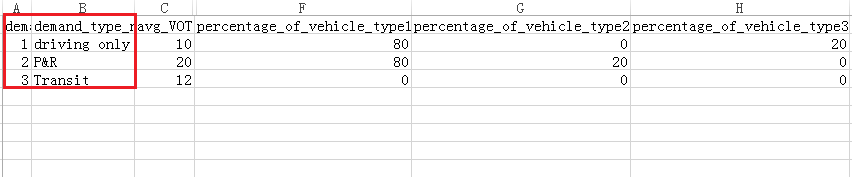
## Intermodal Scenario

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

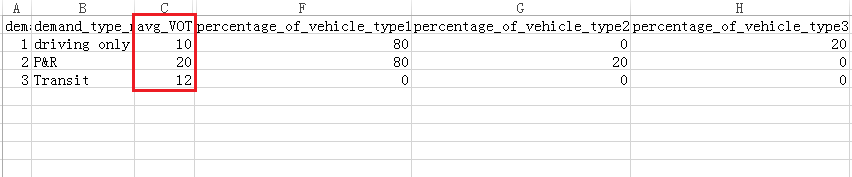
See more details in section **4.1. Basic Traffic Assignment Scenario**.

It should be noted that:

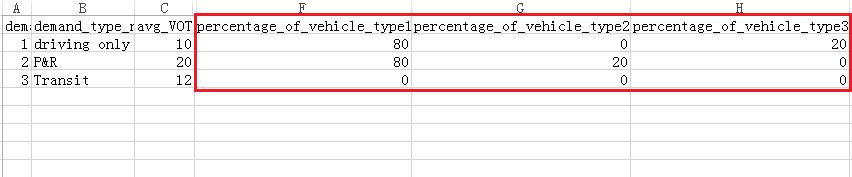
* In file
* Define demand types (e.g., there are 3 demand type such as driving only, park and ride, transit in the example intermodal network)



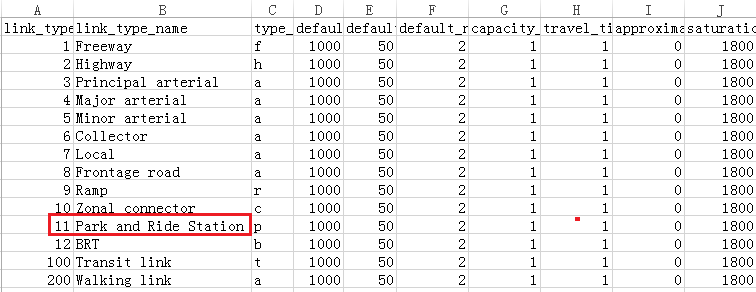
* Set average VOT as the basic of emission data calculation



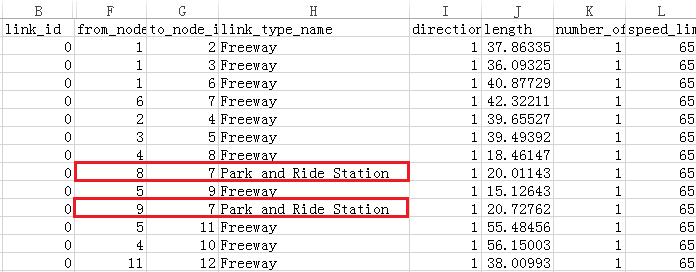
* Specify percentage of different vehicle types



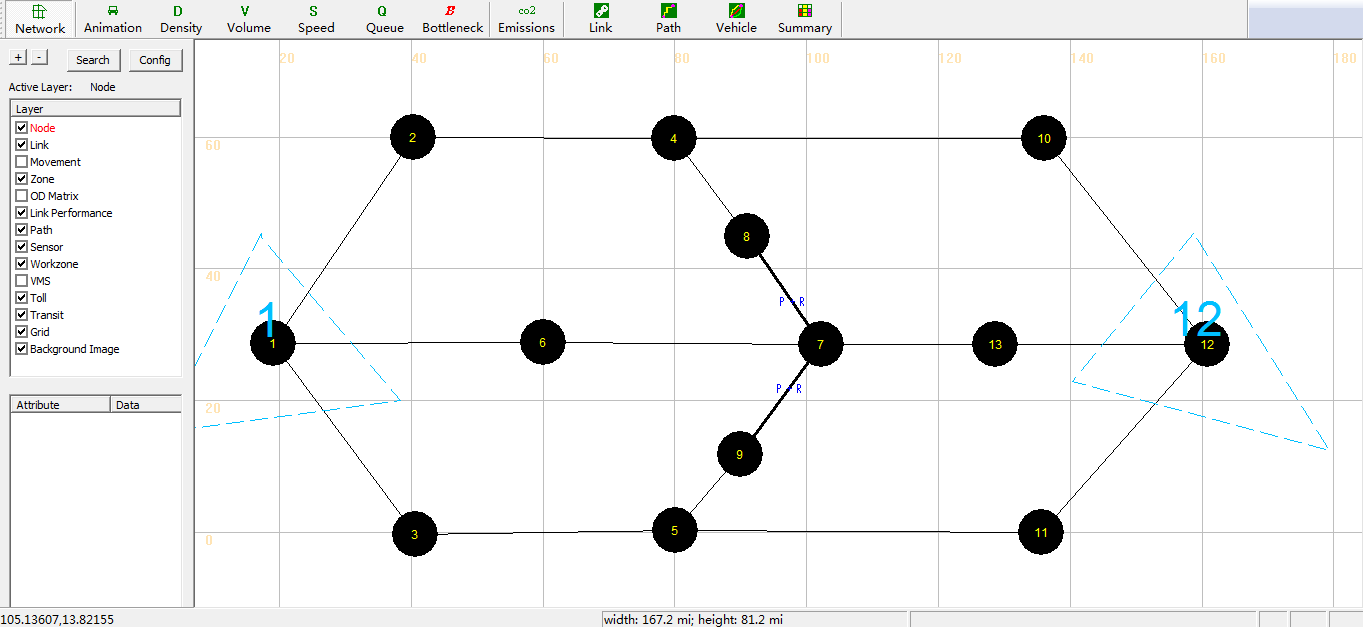
* In file
* Define park-and-ride link type



* In file
* Specify reasonable link type of links (e.g., set link type of road links connected to PR station to “7> Park and Ride Station”)

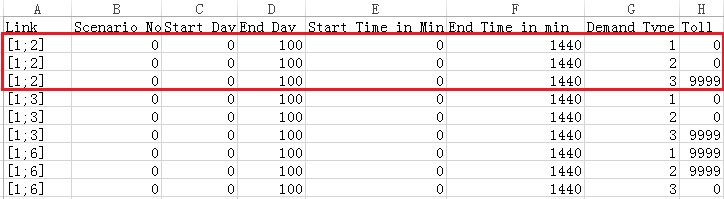


A intermal network display in NeXTA is shown in the figure below

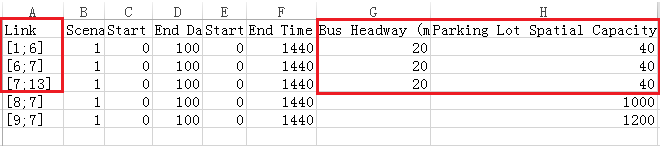


#### Step 2: Prepare Specific Input Files for Intermodal Scenario

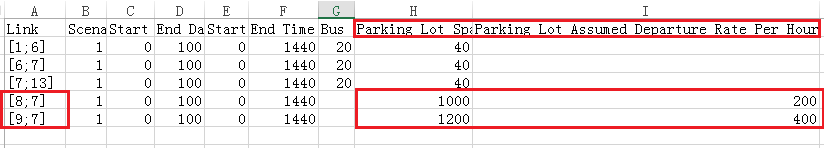
* Step 2-1: Setfile
* Define demand type permitted on different links by specify toll standard. If the demand type is permitted, its toll is 0; otherwise, one big toll value (9999) is put on this link to prohibit this demand type.
* Specify start and end iteration, toll time period of different demand type on links



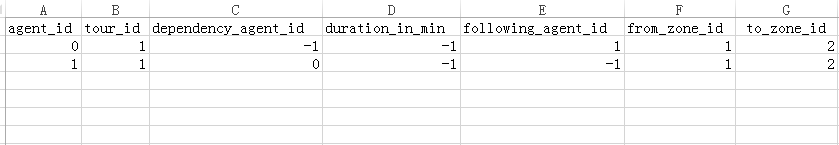
* Step 2-2: Set file
* For transit type links, bus headway and parking lot spatial capacity should be set. It should be noted that “parking lot spatial capacity” means transportation capacity of a bus (usually set to “40”)

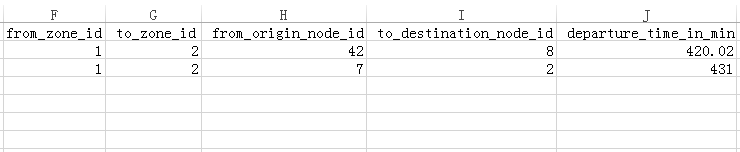


* For parking-and-ride type links, define parking lot spatial capacity and its assumed departure rate



* Step 2-3: Set file





#### Step 3: Run Simulation and Perform Analysis

* Step 3-1: Double-click  after checking all input files, run simulation and perform analysis

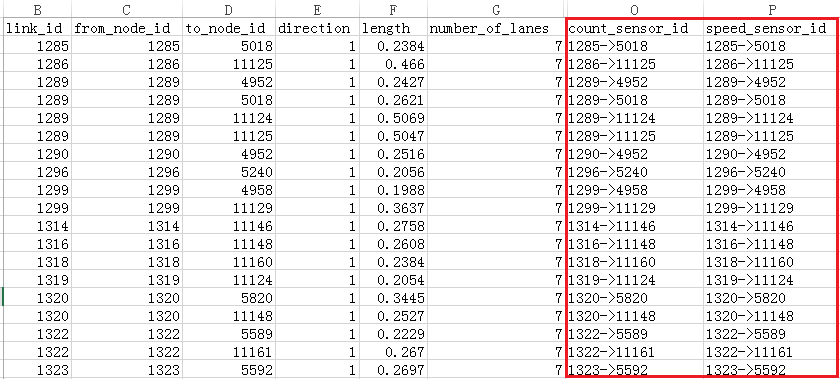
## OD Matrix Estimation

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

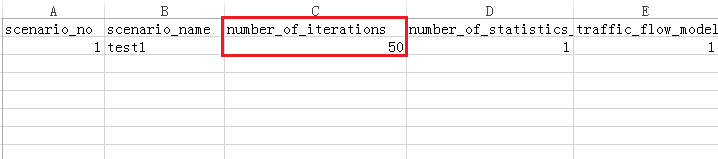
See more details in section **4.1. Basic Traffic Assignment Scenario**.

In addition, It should be noted that:

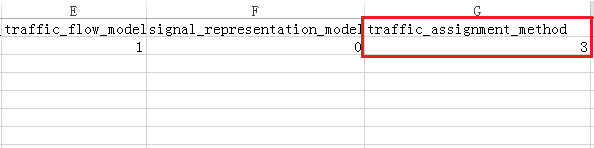
* In file
* Set count sensor ID to “origin node -> destination node”
* Set speed sensor ID to “origin node -> destination node”



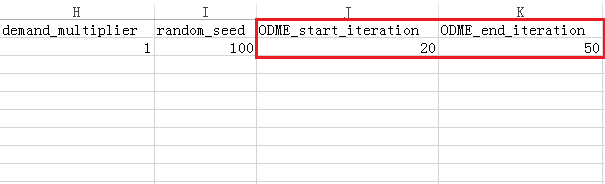
* In file
* To reach a reasonable converge, at least 3-10 global iterations are usually needed, so at least 30 iterations are required for the path flow adjustment for ODME



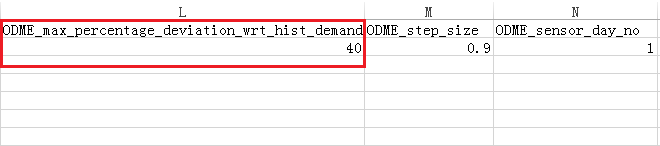
* Specify traffic assignment method to “3>ODME”



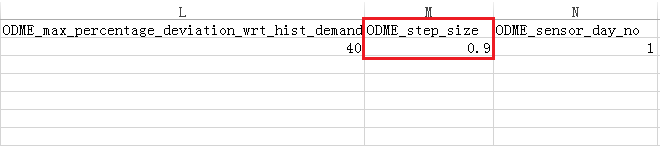
* Specify start and end iteration of ODME (the iteration number indicate that ODME will begin at the 21th iteration and end at the 50th iteration)



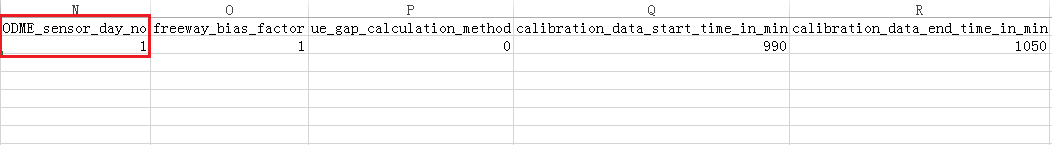
* Specify the maximum percentage of demand deviation from base-line dynamic demand



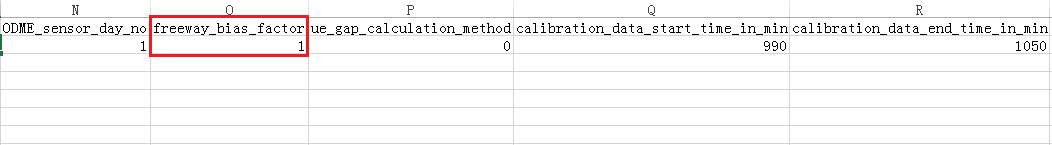
* Specify moving size of each step in path flow adjustment algorithm



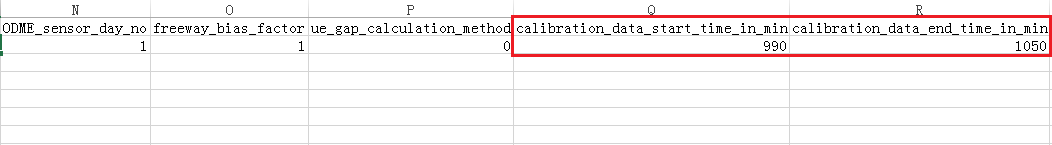
* Specify the sequence number of ODME sensor day you need if there are several days of sensor data



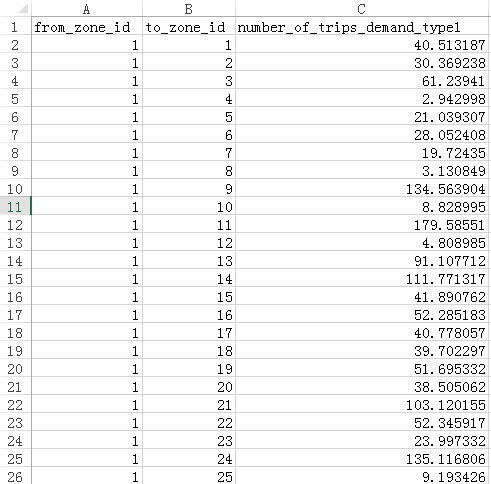
* Specify freeway bias factor



* Specify the time window for ODME to use the sensor data

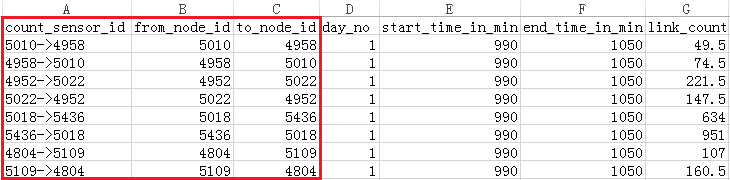


* In file
* A OD demand matrix seed is required as our initial demand values for ODME

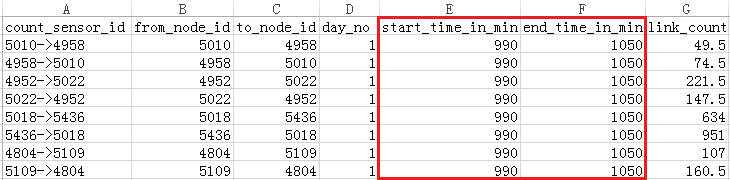


#### Step 2: Prepare Specific Input File for Signal Scenario

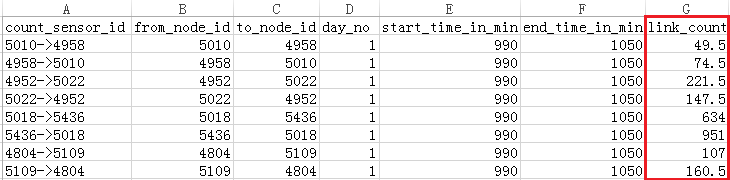
* Step 2-1: Setfile
* Specify count sensor ID, origin node ID and destination node ID of observed links (corresponding observed links infile )



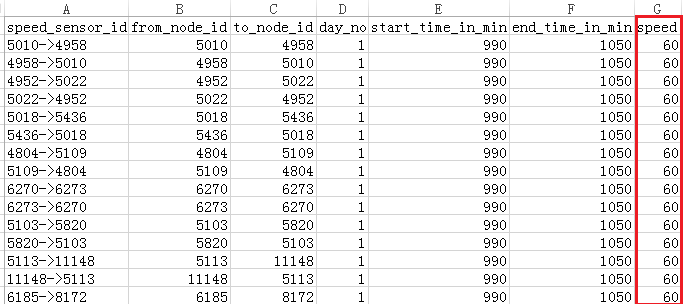
* Specify observation time period of observed links



* Set link count of observed links in specific observation time period

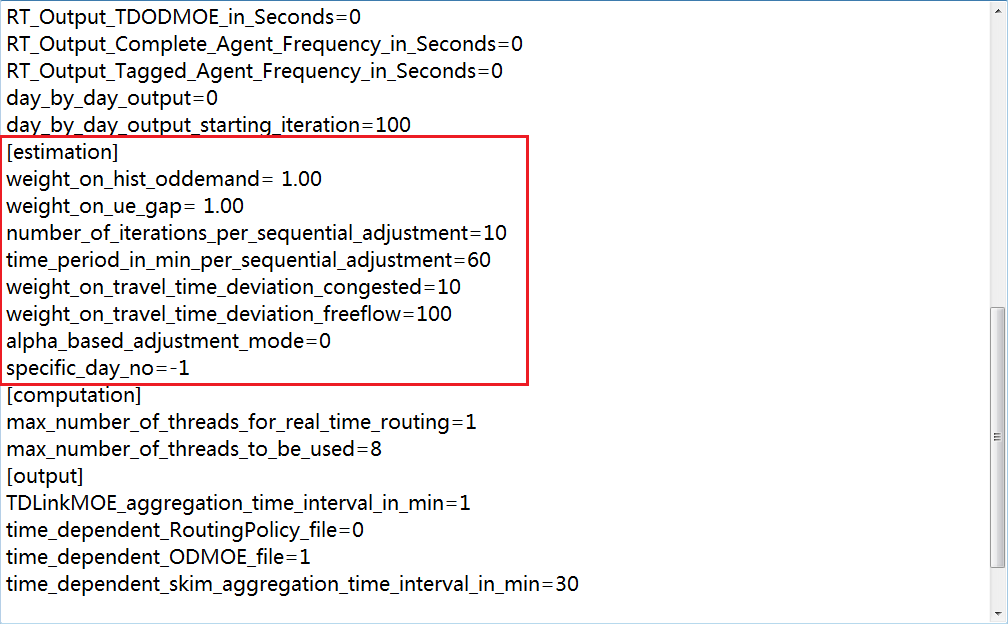


* Step 2-2: Setfile
* Specify speed sensor ID, origin node ID, destination node ID, observation time period of observed links(similar to parameter settings in  file)
* Set observed speed of observed links in specific observation time period

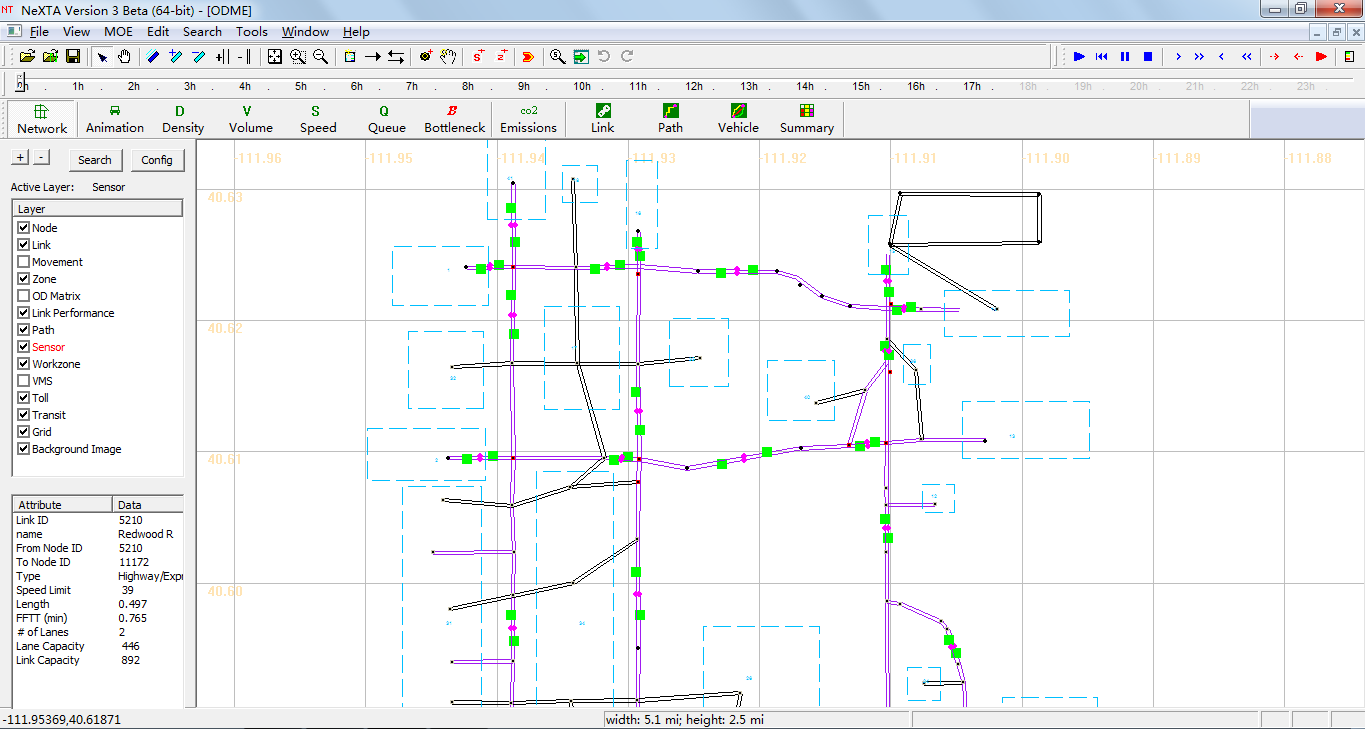


#### Step 3: Run Simulation and Perform Analysis

* Step 3-1: Setfile



* Specify the number of iterations for DTALite to adjust its demand during the time period of one sequential adjustment above (60 min or 1 hour). In this case, the calibrated demand period is 1 hour (according to calibration data start time and end time in to  file), so it requires 10 routing-simulation iterations to complete one global iteration for this demand period.
* Set the time period of each sequential adjustment in the algorithm(e.g., 60min in this case)
* In order to improve the computational efficiency, users can set the value of “max\_number\_of\_threads\_to\_be\_used” to what you want based on your computer configuration to perform parallel computing in DTALite
* The traffic network displayed in NeXTA is shown in the figure below. Those green dots on some links represent the sensors’ location in the network



* Step 3-2: Double-click  after checking all input files, run simulation and perform analysis