|  |
| --- |
| **User Guide for DTALite for GMNS**  **Version 0.9** |

|  |
| --- |
| Revision Date  02-17-2020 |

**CONTACT INFORMATION**

Further Details in <https://github.com/xzhou99/Dtalite_traffic_assignment>

For more information, please contact:

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

**TABLE OF CONTENTS**

[1. GETTING STARTED 2](#_Toc15766)

[1.1 What is GMNS? 2](#_Toc16223)

[1.2 What is AMS? 2](#_Toc681)

[1.3 What is NEXTA? 2](#_Toc8331)

[1.4 What is DTALite? 2](#_Toc2704)

[2. DATA FILE DESCRIPTION 2](#_Toc20661)

[2.1 High-level introduction of data files 2](#_Toc17440)

[2.2 GMNS network input files 3](#_Toc24830)

[2.3 DTALite Definition and Demand Input file 5](#_Toc2801)

[2.4 DTALite Output file 5](#_Toc6559)

[3. Example of Braess's paradox 7](#_Toc23921)

# GETTING STARTED

## What is GMNS?

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

Further Details in <https://zephyrtransport.org/projects/2-network-standard-and-tools/>

## What is AMS?

As stated in FHWA website, <https://cms7.fhwa.dot.gov/research/operations/analysis-modeling-simulation/analysis-modeling-simulation-overview>, FHWA and its State and local agency partners have relied on analysis, modeling, and simulation (AMS) to support investment decisions for the transportation system. As the transportation system environment grows in complexity, increasing pressure is placed on agencies to identify more innovative and efficient solutions to a wide range of issues. These solutions include leveraging emerging technologies, data sources, and alternative (non-traditional) strategies. AMS tools will continue to play a critical role in evaluating these solutions.

## What is NEXTA?

In general, the software suite of 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 environment.

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

This section describes all input and output files associated with NeXTA package. All GMNS 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.

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

## What is DTALite?

Static traffic assignment.

Below is a short list of key features for GMNS data files and simple AMS data structure.

# DATA FILE DESCRIPTION

### High-level introduction of 6 data files

High-level introductions:

* A generic network used for GMNS readable by NeXTA includes a set of three layers: node, link and movement, defined in node.csv and link.csv.
* A link is defined using upstream node and downstream node ids, with essential attributes such as length, free\_speed, lanes and capacity, typically required for static traffic assignment and mesoscopic traffic assignment.
* The node and link layers can use arbitrary coordinate system, but a WKT (lon/lat) coordinate system is preferred.

|  |  |  |
| --- | --- | --- |
| File type | Index: file name | Description |
| Input network file | 1: *node.csv* | Define nodes in the network. |
| 2: *link.csv* | Define links in the network, including parameters used in Volume-Delay Function (VDF) |
| Input settings file (settings.csv) | 3a: *link\_type* | Define types of links in the network. Each type of link could have specific agent types, except which agents of other types can’t use links of this type. |
| 3b: *agent\_type* | Define attributes of each type of agent, including VOT (unit: dollar per hour) |
| Input demand file | 3c: *demand\_file\_list in settings.csv* | Define demand type, period, and agent type. |
| 3d: *demand\_period* | Define demand period, which could be extracted by *demand\_file\_list* |
| 4: *demand.csv* | Define the demand of passengers on each OD pair, which could be extracted by *demand\_file\_list.csv*. |
|  |  |
| Output file | 5: *agent.csv* | Show the results of the assignment, including the volume, cost, travel time and distance of each path of each agent, as well as the link sequence and time sequence. |
| 6: link\_performance.csv | Show the performance of each link, including the travel time, total volume, volume for different agent types,  and resource balance. |

### GMNS network input files

#### 1. node.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| name | Optional for visualization only | Main street @ Highland Dr. |
| node\_id | Node identification number | 1001 |
| ctrl\_type | Intersection control type | 5 |
| node\_type | Optional text label for visualization and identifies of node | 1 |
| x\_coord | Longitude or horizontal coordinate in any arbitrary geographic coordinate system. | 100 |
| y\_coord | Latitude or vertical coordinate horizontal coordinate in any arbitrary geographic coordinate system | 200 |
| geometry | Text string used to describe node location <https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry> | POINT (30 10) |

**Remarks:**

#### 2. link.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| name | Optional for visualization purposes | Main Street |
| link\_id | Link identification number of the road | 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 | Optional text label for visualization and data checking purposes | 1 |
| length | The length of the link (between end nodes), measured in units of miles. | 1.0 |
| lanes | The number of lanes on the link | 2 |
| free\_speed | Free-flow speed on defined link . Suggested Unit: mph or kmph | 20 |
| capacity | The number of vehicles per hour per lane. | 1500 |
| geometry | Text string used to describe link shape and location (typically in WKT 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 (30 10, 10 30, 40 40) |

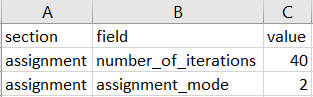
**Remarks:**

**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.

### DTALite Definition and Demand Input Setting File

*settings.csv*



Here we set the number of iterations to be 40. And we set assignment mode to be 2, which means Equilibrium with Penalty is selected as the method to assign.

|  |
| --- |
| 3a: *link\_type* |
| 3b: *agent\_type* |
| 3c: *demand\_file\_list* |
| 3d: *demand\_period.csv* |

### DTALite Output file

High-level introductions:

* Dynamic AMS data visualization files readable by NeXTA includes a set of two layers: agent, link\_performance.
* The specific file names are agent.csv, link\_performance.csv.
* The agent file contains the specific information of each agent in the simulation network, such as, agent id, demand type, time period and so on.
* The link performance file contains the each link’s information, such as, time period, travel time and some notes.

#### 1. agent.csv

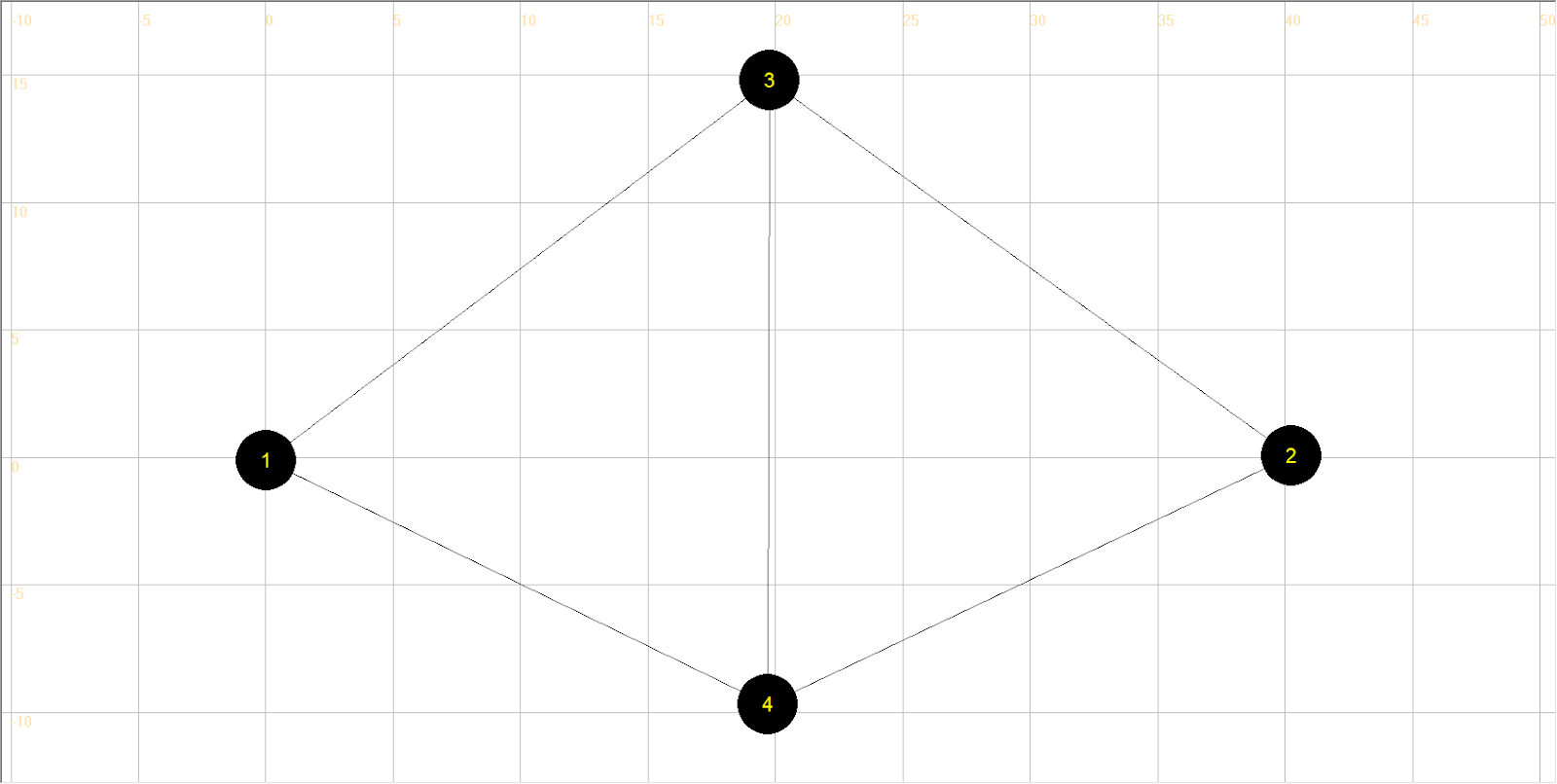
|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| agent\_id | Node identification number | 1 |
| o\_zone\_id | Origin zone number of the agent | 1 |
| d\_zone\_id | Destination zone number of the agent | 7 |
| o\_node\_id | Origin node number of the agent | 1 |
| d\_node\_id | Destination node number of the agent | 20 |
| demand\_type | Optional demand label for visualization and identifies of agent | SOV |
| time\_period | The simulation time period of the agent | 0700\_0900 |
| volume | Maximum flow rate for each lane on the link, in vehicles per hour | 60 |
| cost | The amount of money/time that agent spend | 360 |
| travel\_time | The total time from the origin to the destination of the agent | 360 |
| distance | The total travel distance from the origin to the destination of the agent | 22 |
| node\_sequence | The number of nodes through which agents pass in turn | 1;2;6;8;7;18;20; |

#### 2. link\_performance.csv

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| link\_id | Link identification number of the road | 1 |
| time\_period | The simulation time period of the agent | 0600\_1100 |
| volume | Link based flow volume for the defined time period | 60 |
| travel\_time | Link travel\_time per min | 60 |
| notes | Some explanatory text | period-based |

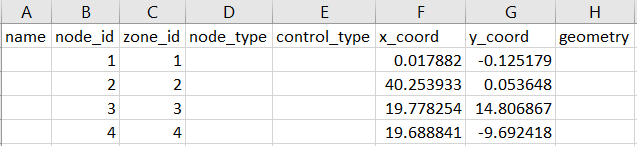
# Example of Braess's paradox

  Braess's paradox is a road network as shown in the figure formed by 4 nodes and 5 links. The travel time of the link from 1 to 3 and the link from 4 to 2 is the number of travelers (T) divided by 100. And the travel time of the link from 1 to 4 and the link from 3 to 2 is a constant 45 minutes. Suppose the link from 3 to 4 is a road with an extremely short travel time of approximately 0 minutes.

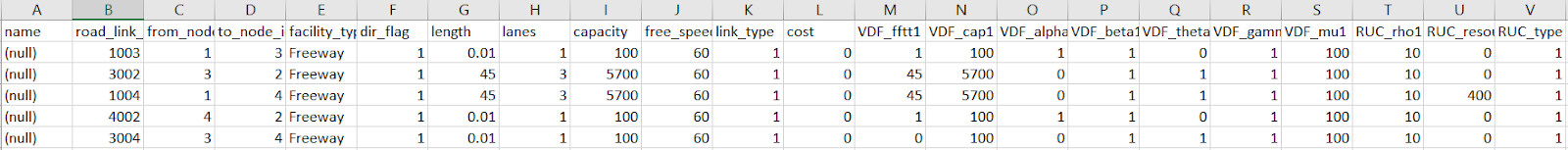


Input network files:

1a) *node.csv*



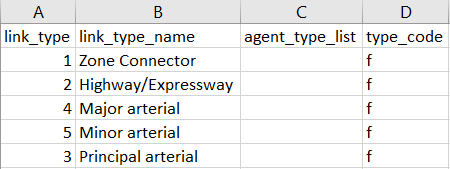
1b) *link.csv*



Here we define the resource base is 400 on the link from 1 to 4. The coefficients of VDF define the travel time on each link as described above.

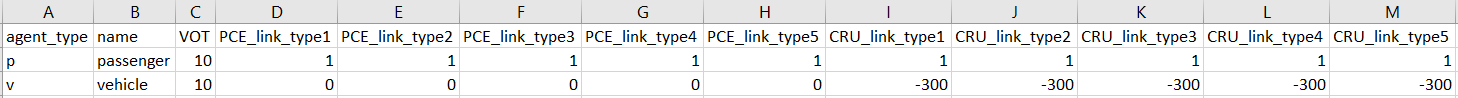
Input settings file:

2a) *link\_type*



Here we leave the column of agent\_type\_list blank, which means all types of agents can use all types of links.

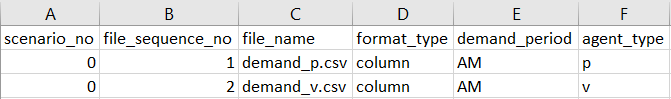
2b) agent*\_type*



Here we define the VOT, PCE of each link type and CRU of each link type for each commodity. We only consider two types of commodities in this case, i.e. passenger and vehicle.

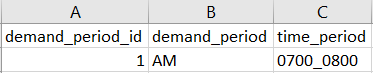
Input demand file:

3b) *demand\_file\_list*



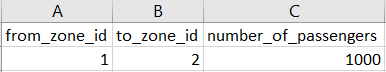
In this case, we only put two types of commodities. So we only need to extract demand data from two files (*demand\_p.csv* and *demand\_v.csv*).

3d) *demand\_period.csv*



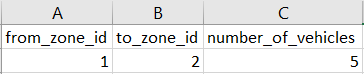
Here we only consider one time period. The departure time is between 7:00am and 8:00am.

3c) *demand.csv*



It is assumed the passenger demand from node 1 to node 2 is 1000 during the time period.

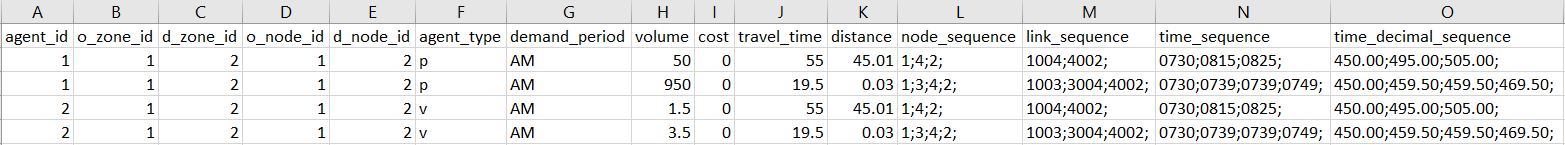
3d) *demand.csv*



It is assumed there are 5 vehicles available travelling from node 1 to node 2 during the time period.

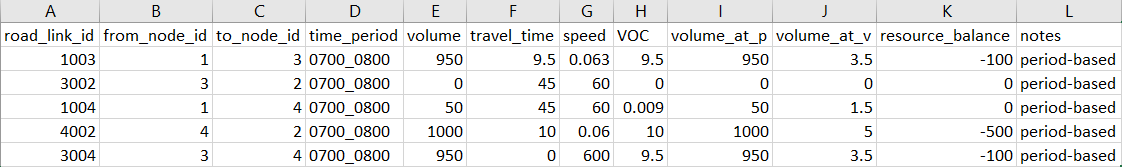
Output files:

5) *agent.csv*

**

Based on the assignment method, total passengers are assigned to two paths: 50 passengers travel through 1-4-2, and 950 passengers travel through 1-3-4-2. Links sequence and time sequence are displayed as well.

6) *link\_performance.csv*

**

The performance of each link is indicated by Travel time, Volume of Capacity, Volume of each commodity and Resource balance. According to the resource balance, we can arrange the supply of vehicles in *demand\_v.csv.*

-

Column generation mode. Mode = 4:

Experiment 1

Experiment 2