# DTALite Users Guide

Working Document Version 1.0

Please feel free to send any questions, feedback, and corrections to Dr. Xuesong (Simon) Zhou (xzhou74@asu.edu) by adding comments in this document.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in [www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)

Table of Contents

[DTALite Users Guide 1](#_Toc110979246)

[1. Introduction 2](#_Toc110979247)

[1.1. Motivation 2](#_Toc110979248)

[1.2 System Architecture 4](#_Toc110979249)

[1.3. 5 steps of performing traffic analysis using CSV files 6](#_Toc110979250)

[2. Getting Started from NeXTA graphical user interface and running DTALite 7](#_Toc110979251)

[Step 1: Download and locate the project folder, check CSV network files. 7](#_Toc110979252)

[Step 2: Visualize and validate network in NeXTA using shortest path finding 8](#_Toc110979253)

[Step 3. Run DTALite as a Windows console application from File Explorer 10](#_Toc110979254)

[3. Toy Examples for Computing Static User Equilibrium 11](#_Toc110979255)

[3.1 Two-corridor example 11](#_Toc110979256)

[3.2 Detailed data structure description 12](#_Toc110979257)

[4. Detailed data structure descriptions 15](#_Toc110979258)

[4.1 Input for network data 15](#_Toc110979259)

[4.2 Input for demand data 17](#_Toc110979260)

[4.3 Assignment and simulation configuration file 17](#_Toc110979261)

[4.4 Input for supply data 19](#_Toc110979262)

[4.5 Input for signal timing and service layer 19](#_Toc110979263)

[4.6 Output file 20](#_Toc110979264)

[Appendix A: From mathematical modeling to network-based assignment and simulation 22](#_Toc110979265)

[Appendix B: The method of preparing the zone ids 24](#_Toc110979266)

# Introduction

## 1.1. Motivation

Motivated by a wide range of transportation network analysis needs, static traffic assignment (STA) and dynamic traffic assignment (DTA) models have been increasingly recognized as a set of important tools for assessing operational performances of those applications at different spatial resolutions (e.g., network, corridor and individual segment levels) and across various analysis temporal regimes (e.g., peak hours, entire day and second-by-second). The mathematical modeling and related volume-delay functions are described in Appendix.

The advances of STA and DTA are built upon the capabilities of integrated flow assignment and simulation models in describing the formation, propagation, and dissipation of traffic congestion in a transportation network.

As a continuation of DTALite, the development of DTALite (S stands for strategic or static assignment) is motivated by the following perspectives.

1. **Bridging the gap from macroscopic static assignment to mesoscopic dynamic assignment**

Planning practitioners have recognized the full potential of DTA modeling methodologies that describe the propagation and dissipation of system congestion with time-dependent trip demands in a transportation network. In April 2009, the TRB Network Modeling Committee conducted a DTA user survey through the FHWA TMIP mail list, which identified the following top 5 technical barriers:

* DTA requires more data than are available or accessible to most users (47%)
* Setting up a DTA model consumed inordinate resource (44%)
* Cost/benefit of implementation is unclear (45%)
* DTA tools take too long to run (35%)
* The underlying modeling approaches are not transparent (35%)

The development goal of DTALite aims to provide an integrated open-source package for strategic traffic analysis that includes both static traffic assignment and dynamic traffic simulation to reflect the impact of road capacity constraints. The underlying volume-delay models include BPR functions and its extension of BPR-X. Three traffic stream models, namely, point queue model, spatial queue model and simplified kinematic wave models, are embedded in the mesoscopic simulator to describe queueing behavior at bottlenecks with tight capacity constraints.

1. **Adopting open network standard of GMNS**

General Travel Network Format Specification is a product of Zephyr Foundation, which aims to advance the field through flexible and efficient support, education, guidance, encouragement, and incubation. Further details can be found in <https://zephyrtransport.org/projects/2-network-standard-and-tools/>

1. **Integrated graphic user interface and analysis package**

NeXTA (Network eXplorer for Traffic Analysis) is another open-source graphic user interface (GUI) for transportation network analysis, while the lower-case “*e*” stands for education with broader impacts. With both open-source traffic assignment/simulation engine (as a simple Windows console application) and graphic user interface, the software suite of DTALite + NeXTA aims to

* provide an open-source code base to enable transportation researchers and software developers to expand its range of Strategic Traffic Assignment capabilities to various traffic management analysis applications.
* present results to other users by visualizing traffic flow dynamics and traveler route choice behavior in an integrated 2D environment.
* provide a free education tool for students to understand the complex decision-making process in transportation planning and optimization processes.

1. **parallel computing on shared memory multi-core computer**

Emerging multi-core computer processor techniques are offering unprecedented available parallel computing power, on most of laptops and desktops currently available in the market. To exploit this paradigm change in computing, we will require a new software architecture and algorithm design so as to facilitate the most efficient use of emergent parallel hardware.

1. **Integrated signal timing optimization (to be added)**
2. **Integrated OD demand estimation through path flow estimator (to be added)**

The latest software release can be downloaded at our Github website. The source code can be downloaded at https://github.com/asu-trans-ai-lab/DTALite. Table 1 illustrates the contents of different folders at Github https://github.com/asu-trans-ai-lab/DTALite.

Table 1. contents of folders at Github.

|  |  |
| --- | --- |
| **Github Folder Name** | **Contents** |
| Src | source code of DTALite |
| test | a simple working dataset for console application DTALite and visualizer of NeXTA. |
| Doc | user’s guide and other documentations for DTALite |
| Data | sample datasets for DTALite:  1. two\_corridor  2. Braess’s\_paradox  3. three\_corridor  4. Sious\_Falls  5. Chicago\_sketch  6. Tempe ASU network |

## 1.2 System Architecture

The software architecture of DTALite aims to integrate many rich modeling and visualization capabilities into an open-source traffic assignment model suitable for practical everyday use within the context of an entire large-scale metropolitan area network. Using a modularized design, the open-source suite of **simulation engine + 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 can allow state DOTs and regional MPOs to rapidly apply the advanced STA/DTA methodology, and further examine the effectiveness of traffic mobility, reliability and safety improvement strategies, individually and in combination, for a large-scale regional network, a subarea or a corridor.



Figure 1.1 Software System Architecture

The components and different modules in the system are listed as following:

**a. Network Data** includes two essential files, node.csv and link.csv for the macroscopic network representation.

**b. OD Demand Meta Database** includes the setting.csv as the configuration file that describes information such as agent type, demand period, demand file list, which help users to represent the OD demand information for different user types at specific demand periods.

**c. Traffic Assignment Module** includes the key steps of the assignment, including the BPR Volume Delay Function, Shortest Path Tree Generation, and Flow Assignment, which generates the path flow and link flow according to the UE principle.

**d. NEXTA: Visualization Interface Module** is able to visualize the network and the output of traffic assignment, including Static Link Performance and Agent Trajectory.

**e. Space-Time Simulation Module** utilizes the path flow output of Traffic Assignment Module to perform Space-Time Simulation, while the underlying traffic flow models in the Space-Time Simulation Module are Point Queue (PQ) and Spatial Queue (SQ). A simplified kinematic wave (KW) model can be also used in an advanced mode, similar to DTALite.

**f. Capacity Management** aims to manage the static and time-dependent link capacity input for Space-Time Simulation, such as signal timing plans and multi-modal service plans.

**g. Simulation Output Module** covers the output file of Space-Time Simulation Module, including Dynamic Link Performance and Agent Trajectory in terms of link\_performance.csv and agent.csv, which can be visualized in NeXTA.

Regarding parameters in settings.csv, Table 2 illustrates the differences between two key steps of Static Traffic Assignment and DTA + space-time simulation.

Table 2. The between Analytical Traffic Assignment and DTA+ network based

|  |  |  |
| --- | --- | --- |
|  | Analytical Dynamic Traffic Assignment | Simulation-based Network loading |
| Travel time evaluation | BPR function with volume/capacity ratio (soft capacity constraints) | Space-time network based simulation with tight capacity constraints |
| Demand input | OD demand | OD demand or agent based input from the analytical DTA |
| Output (1):  link performance | Link performance, Dynamic Link performance based on  Queue VDF |  |
| Output (2): route\_assignment | Route\_assignment  Node\_sequence, link\_sequence |  |
| Output (3):  trajectory | no trajectory.csv with static traffic assignment | Individual agent trajectory with path sequence and time sequence |

See [DLSim-MRM/settings.csv at main · asu-trans-ai-lab/DLSim-MRM (github.com)](https://github.com/asu-trans-ai-lab/DLSim-MRM/blob/main/dataset/4_101_corridor/settings.csv) for more details

OD estimation

Input: measurement.csv. output: adjusted path volume in route\_assignment.csv and link volume (before and after odme) in link\_performance.csv

Focus and O step: input: subarea.csv. node and link, and demand, output subarea\_related\_zones.csv that show the mapping between original zone id and super\_zone id.

Origin zone based aggregation.

For all zones inside the subarea.csv. will be kept, we will aggregate different TAZ to super zones. route assignment is based on super zone structure.

F:

O

C: columnn generation

U: columna updating with the key performance as UE gap.

S: sensitivity analysis: input is supply\_side\_scenarion.csv. output: SA before and after volume, and travel time in link\_performance and route\_assignment.csv

I: information class:

N

G

## 1.3. 5 steps of performing traffic analysis using CSV files

The specific instruction for the use of NeXTA and DTALite is as follows:

Step 0: **[Download and locate the project folder]** Download and unzip the release software package from github. Locate DTALite file folder with node.csv, link csv, demand.csv and settings.csv. Typically, copy DTALite.exe and NeXTA.exe in the same folder for easy access.

Step 1: **[Check input files in Excel]** Open a file explorer, view or edit input files of node, link and demand csv files, in Excel or any text editor.

Pay attention to the node.csv, check if there are zone\_ids in the file, if not, you need to complete it. See the Appendix B for the method of the method of **preparing the zone ids**.

Review and change the configuration in settings.csv in Excel.

Step 2: **[Visualize and validate network in NeXTA]** Click “NeXTA”—“File”—“Open Traffic Network Project” to choose the node.csv file in your network data set. Check the network connectivity through a simple path calculation by selecting one OD pair.

Step 3: **[Run DTALite as a Windows console application]** Click on the executable of “DTALite.exe” from a file explorer or run it from Windows command window, to perform traffic assignment and simulation. The output of this Windows console applications is displayed in screen and log file DTALite\_log.txt.

Step 4: [**Check output files in Excel**] After the completion of DTALite, users can view the output link performance and agent files in Excel.

Step 5: [**Visualize output files in NeXTA**] For static traffic assignment, NeXTA is able to display view link travel time, speed and volume, as well as path display in the agent dialog. For dynamic assignment and simulation, one can use NeXTA to view time-dependent queue and density.

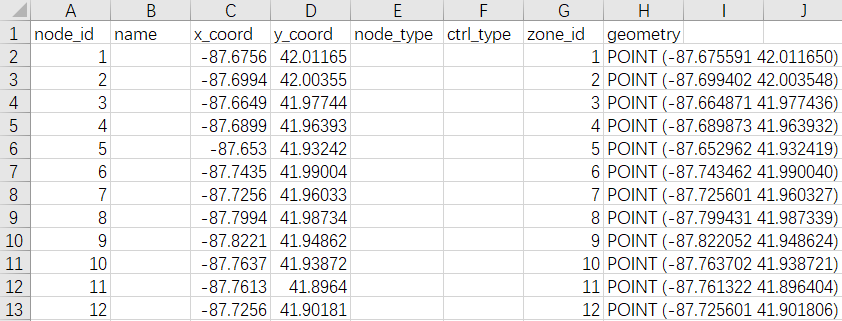
# Getting Started from NeXTA graphical user interface and running DTALite

## Step 1: Download and locate the project folder, check CSV network files.

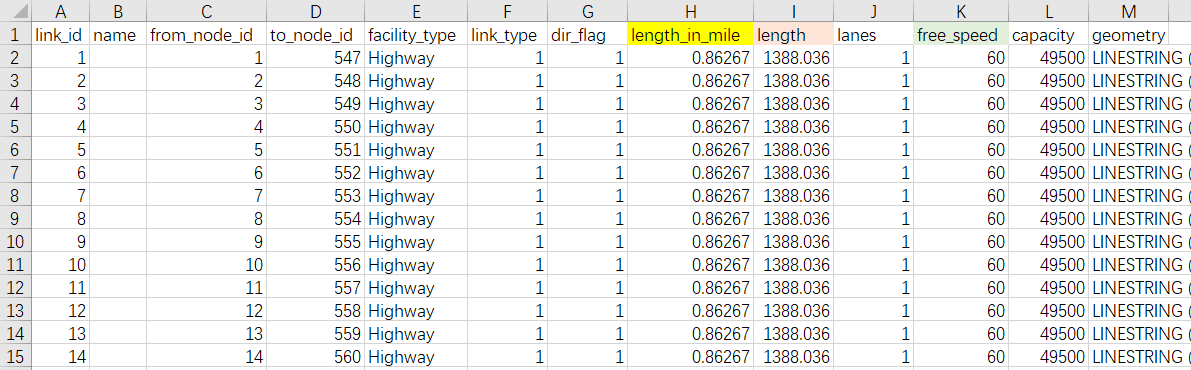
Locate the project folder of “6\_Tempe ASU network”.

Check input files in Excel

node.csv



link.csv



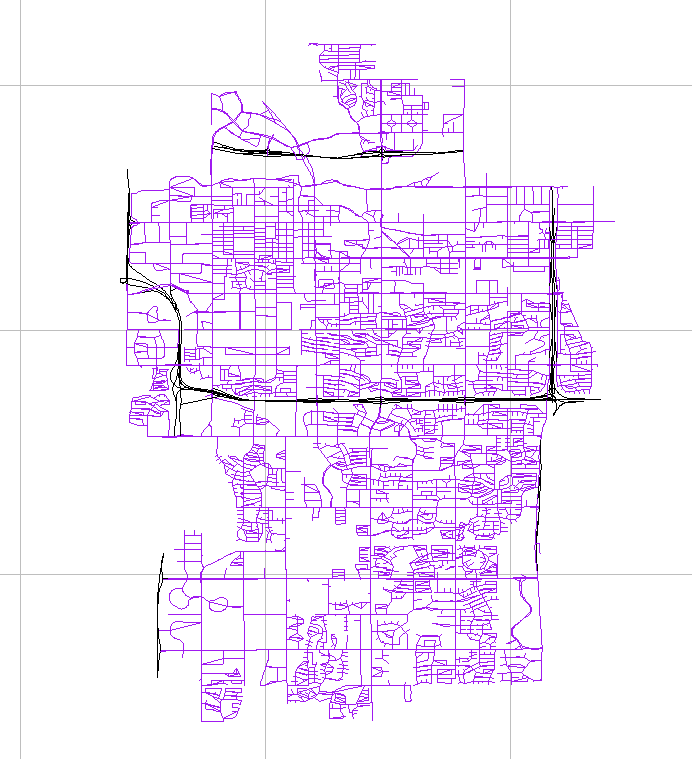
Pay attention to the units in the new version DTALite:

Unit of **length**: meter

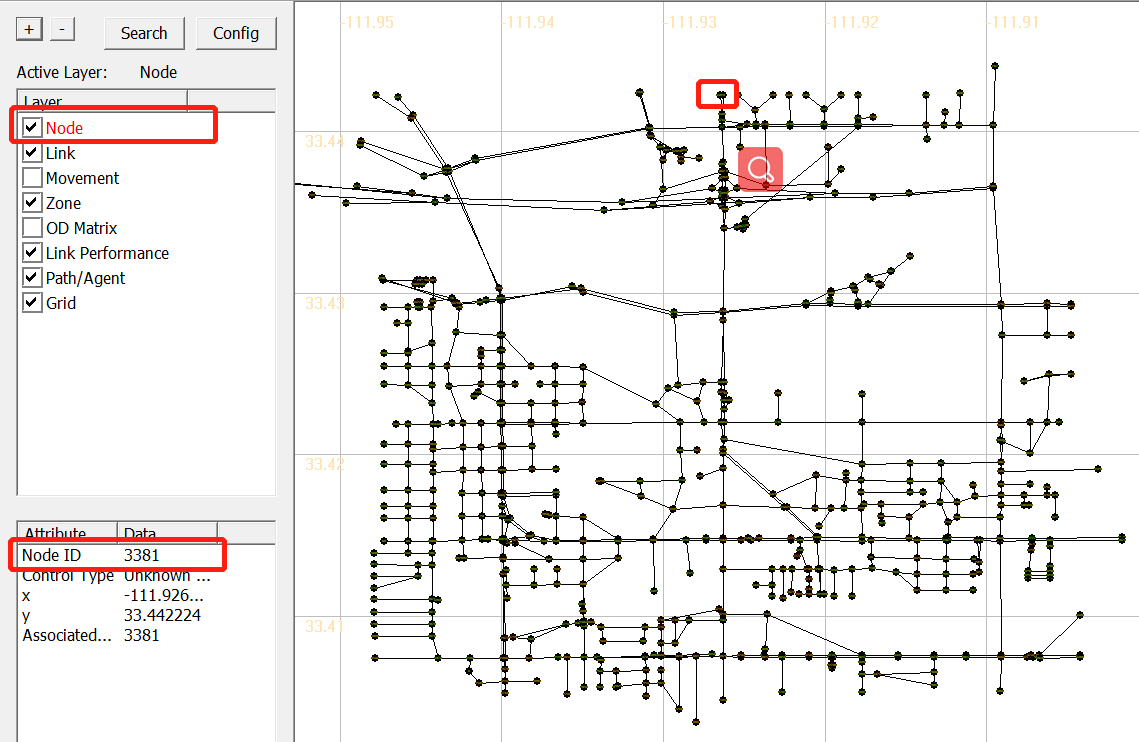
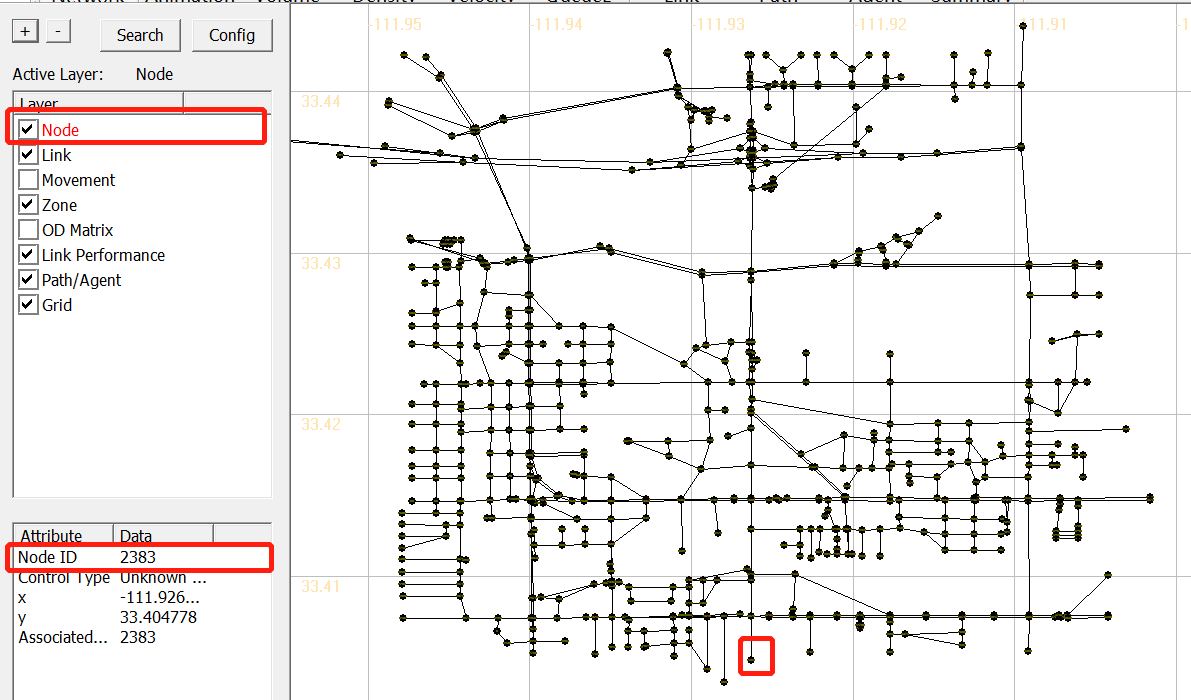
Unit of **free\_speed**: mile per hour (mph)

Unit of **VDF\_fftt1**: min

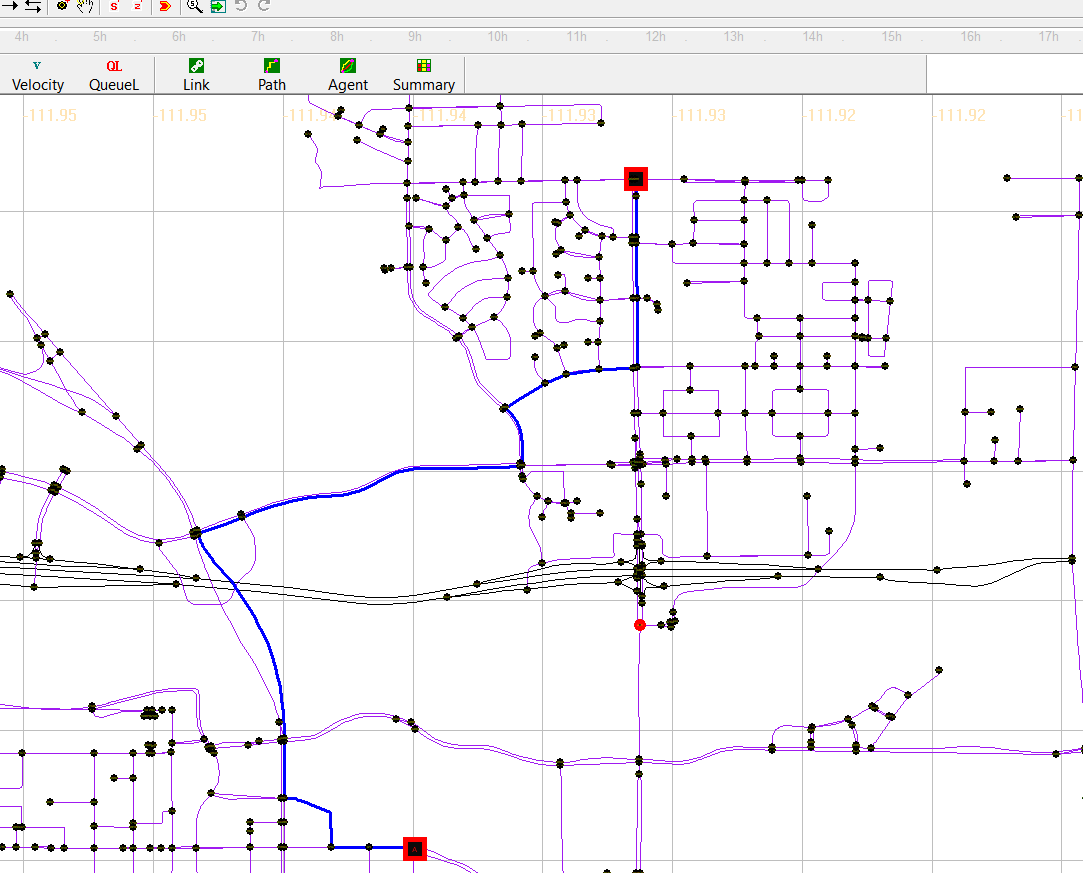
## Step 2: Visualize and validate network in NeXTA using shortest path finding



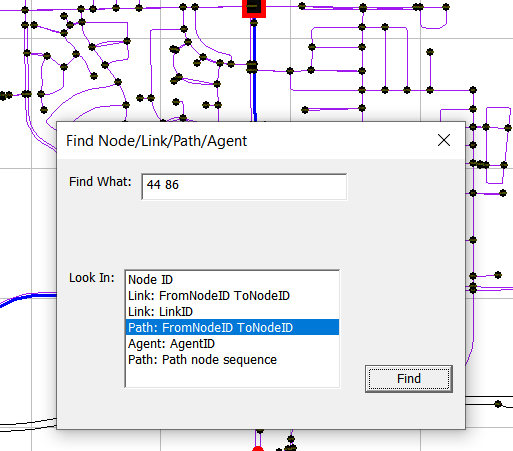
First, select the node layer in the left-hand-side GIS panel, we can use the mouse to select node 44, and node 86. Alternatively, one can use a keyboard shortcut of Control+f to search those nodes.



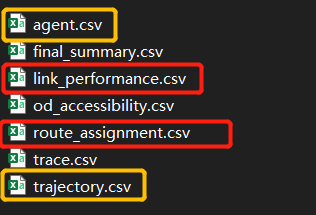
Go the path GIS layer, right click to check and confirm if this path is connected.



Alternatively, one can use a keyboard shortcut of Control+f to specify the origin and destination for the path.



## Step 3. Run DTALite as a Windows console application from File Explorer



The user now can check output files in Excel for the following two files:

1. link performance.csv
2. route\_assignment.csv

If the user also performs a dynamic simulation, the following two files will be generated in addition to the above two files after simulation:

1. agent.csv
2. trajectory

All DTALite **data files** are in CSV format. The files for node, link and zone layers have geometric fields for importing from and exporting to GIS software.

# Toy Examples for Computing Static User Equilibrium

## 3.1 Two-corridor example

This example uses a simple case with a single origin-to-destination pair and two paths p=1 for the primary path, p=2 for the alternative path, see in Figure 3.1 As each path has two links, path 1 has a free-flow travel time of 20 minutes, and path 2 has a free-flow travel time of 30 minutes.

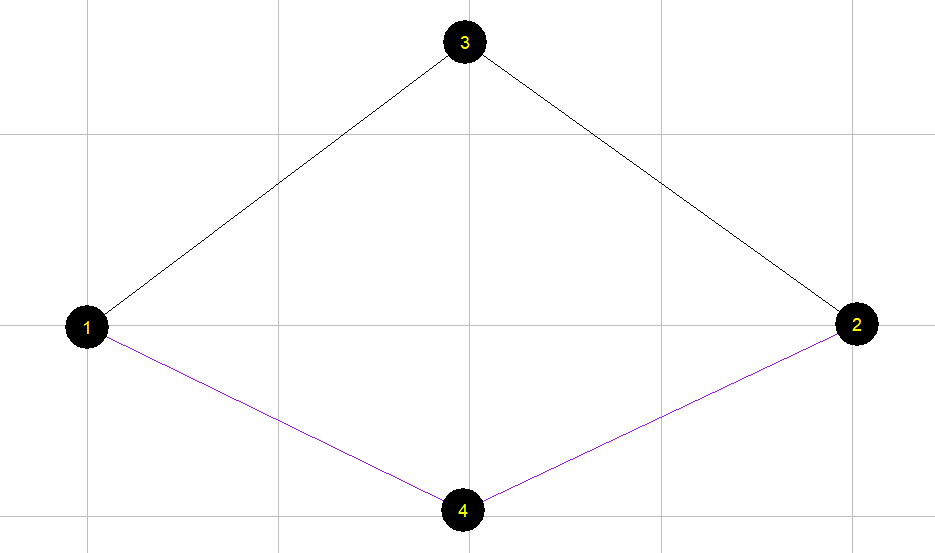


Figure 3.1 illustrative example of two-corridor network

For a given OD demand of 7,000 on this network, we can use the User Equilibrium method to perform traffic assignment. A graphic-based solution process can be described by Figure 3.2. As the path flow changes, the travel time on the two paths reaches the same equilibrium point, which satisfied the requirement of User Equilibrium. User equilibrium solution is reached when the freeway flow is 5400, and arterial flow as 7000-5400=1600, and this leads to the same travel time of 30 min on both routes.

Figure 3.2 illustration of Equilibrium with X axis as freeway path flow.

The detailed parameters are in Table 3.1.

Table 3.1 parameters

|  |  |
| --- | --- |
| **Parameters** | **Value** |
| Freeway flow travel time (min): Freeway: | 20 |
| Freeway flow travel time (min): Arterial: | 30 |
| Capacity (vehicles / hour): Freeway: | 4000 |
| Capacity (vehicles / hour):Arterial: | 3000 |
| Demand | 7000 |
| BPR alpha | 0.15 |
| BPR beta | 4 |

The travel time function is

Freeway\_TT = FFTT[1 + 0.15(v/c)4]

Arterial \_TT= FFTT[1 + 0.15((demand-v)/c)4]

where:

TT = link travel time

FFTT= free-flow travel time of link

v = link flow

c = link capacity

## 3.2 Detailed data structure description

Generic network files used for DTALite include files for three layers: physical layer, service layer and demand layer.

**Table 3.1 File list for DTALite**

|  |  |  |
| --- | --- | --- |
| File type | Index: file name | Description |
| Input for physical layer | 1a: *node.csv* | Define nodes in the network. |
| 1b.: *link.csv* | Define links in the network with essential attributes for assignment. |
| Input for demand layer | 2: *demand.csv* | Define the demand of passengers on each OD pair, which could be extracted by *demand\_file\_list.csv*. |
| Input configuration file | 3: *settings.csv* | Define basic setting for the Network, it contains five sections. |
|  | Section of assignment | Set the number of iteration. |
|  | Section of agent\_type | Define attributes of each type of agent, including vot (unit: dollar per hour) and pce. |
|  | Section of link\_type | Define types of links in the network |
|  | Section of demand\_period | Define demand period, which could be extracted by demand\_file\_list |
|  | Section of demand\_file\_list | Define demand type, period, and format type. |
| Output file | 4: od\_accessibility.csv | Show the accessibility of each OD, each line represents an OD, including the travel time, agent type, and the information of the od zones. |
| 5a: *link\_performance.csv* | Show the performance of each link, including the travel time, volume, and resource balance. |
| 5b: *route\_assignment.csv* | Show the results of the assignment, including the volume, toll, travel time and distance of each path of each agent, as well as the link sequence and time sequence. |
| *6a: agent.cv*  *6b:trajectory* | Show the results of the individual agent trajectory with path sequence and time sequence |

The related files used in DTALite are listed below.

**（1）Prepare input data**

* node.csv

Table 3.2 node.csv

|  |  |  |  |
| --- | --- | --- | --- |
| **node\_id** | **zone\_id** | **x\_coord** | **y\_coord** |
| 1 | 1 | 0.017882 | -0.12518 |
| 2 | 2 | 40.25393 | 0.053648 |
| 3 |  | 19.77825 | 14.80687 |
| 4 |  | 19.68884 | -9.69242 |

* link.csv

Table 3.3 link.csv

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **link\_id** | **from\_node\_id** | **to\_node\_id** | **facility\_type** | **dir\_flag** | **Length(meter)** | **lanes** |
| 1003 | 1 | 3 | Freeway | 1 | 10 | 1 |
| 3002 | 3 | 2 | Freeway | 1 | 10 | 1 |
| 1004 | 1 | 4 | arterial | 1 | 15 | 1 |
| 4002 | 4 | 2 | arterial | 1 | 15 | 1 |
| **capacity** | **free\_speed** | **link\_type** | **VDF\_fftt1** | **VDF\_cap1** | **VDF\_alpha1** | **VDF\_beta1** |
| 4000 | 60 | 1 | 20 | 4000 | 0.15 | 4 |
| 4000 | 60 | 1 | 0 | 4000 | 0.15 | 4 |
| 3000 | 60 | 2 | 30 | 3000 | 0.15 | 4 |
| 3000 | 60 | 2 | 30 | 3000 | 0.15 | 4 |

* demand.csv

Table 3.6 demand.csv

|  |  |  |
| --- | --- | --- |
| **o\_zone\_id** | **d\_zone\_id** | **volume** |
| 1 | 2 | 7000 |

* settings.csv

There are different sections in the settings.csv file. And each section starts with the format of [section\_name] along with the field names. There are five sections in the settings.csv, see in Table 3.4.

**Calendar

Description automatically generated**

**(2) Check output files**

The files are the output of the previous input data.

* agent.csv

Table 3.11 link\_performance.csv

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **link\_id** | **from\_node\_id** | **to\_node\_id** | **time\_period** | **volume** | **travel\_time** | **speed** | **VOC** | **notes** |
| 1003 | 1 | 3 | 0700\_0800 | 5447.848 | 30.322 | 39.575 | 1.362 | period-based |
| 3002 | 3 | 2 | 0700\_0800 | 5447.848 | 0 | 0 | 1.362 | period-based |
| 1004 | 1 | 4 | 0700\_0800 | 1552.149 | 30.322 | 59.362 | 0.517 | period-based |
| 4002 | 4 | 2 | 0700\_0800 | 1552.149 | 0 | 0 | 0.517 | period-based |

From link\_performance.csv, users are able to obtain the link volume, link travel\_time, speed and VOC. For the above example (Table 3.11), the first link id is 1003, and the from-node of this link is 1, and the end-node of this link is 3. From 7:00 to 8:00, the volume on this link is 5447.848, and the travel time during this period is 30.322 (min), and the speed is 39.575 mile per hour, and the volume over capacity (VOC) is 1.362.

Table 3.10 agent.csv

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **agent\_id** | **o\_zone\_id** | **d\_zone\_id** | **path\_id** | **o\_node\_id** | **d\_node\_id** |
| 1 | 1 | 2 | 0 | 1 | 2 |
| 2 | 1 | 2 | 1 | 1 | 2 |
| **agent\_type** | **demand\_period** | **agent\_type** | **toll** | **travel\_time** | **distance** |
| p | AM | p | 0 | 30.3224 | 20 |
| p | AM | p | 0 | 30.3224 | 30 |
|  | **node\_sequence** | **link\_sequence** | **time\_sequence** |  |  |
|  | 1;3;2; | 1003;3002; | 0730:00;0800:19;0800:19; |  |  |
|  | 1;4;2; | 1004;4002; | 0800:19;0830:38;0830:38; |  |  |

The volume in this file represents path volume, and the path is further represented in node\_sequence.

For the above example (Table 3.10), when the assignment reaches equilibrium, there are two paths to choose. For path id 0, the origin zone id is 1 and destination zone id is 2, and the node sequence of this path is 1, and travel time on this path is 30.3224 minutes, and distance of this path is 20.

**（3）Visualize the output in NEXTA**

Open NEXTA, import the network, chose the time period that you set in demand\_period.csv, and click volume, you can see the assignment outcome in Figure 3.3.

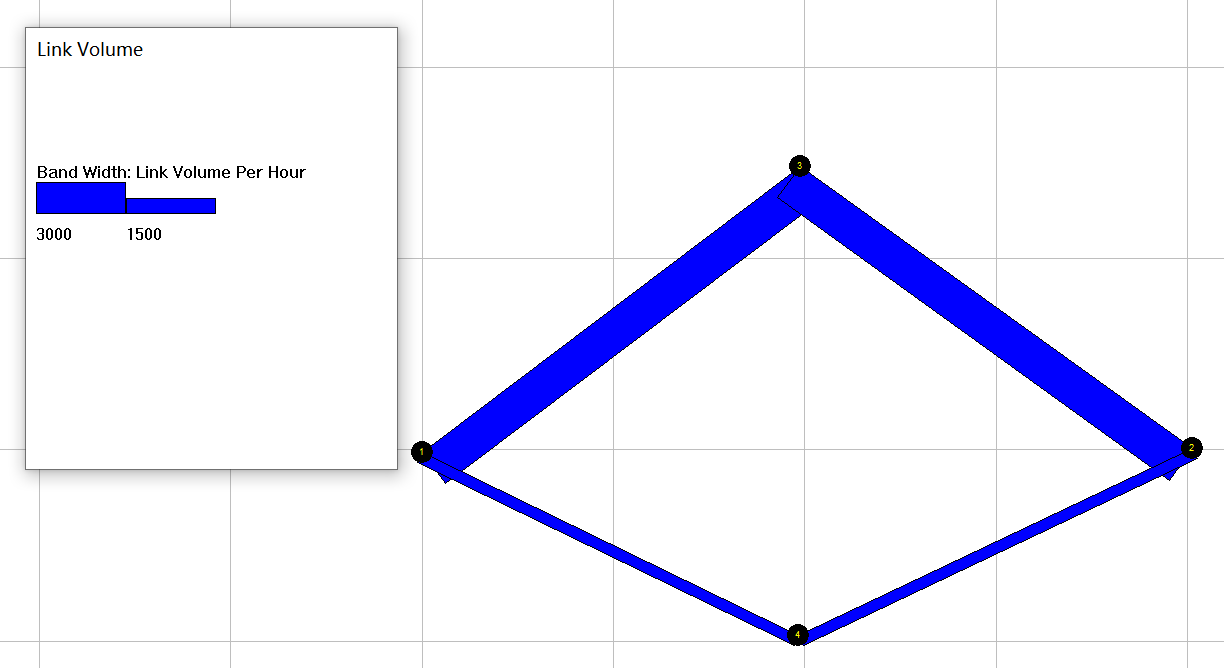


Figure 3.3 Link volume visualization

# 4. Detailed data structure descriptions

## 4.1 Input for network data

* The specific files for physical layer are *node.csv* and *link.csv*.
* Nodes in the physical network represent points of demand, including node\_id, zone\_id, and coordinates with an arbitrary coordinate system.
* A link is defined using upstream node and downstream node ids, with essential attributes such as length, free\_speed, lanes, capacity, link\_type, and coefficients of Volume Delay Function, typically required for static traffic assignment and mesoscopic traffic assignment.

**File 1a: node.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Value** |
| name | Optional for visualization only | Main street @ Highland Dr. |
| node\_id | Node identification number | 1001 |
| 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 |
| node\_type | Optional text label for visualization and identifies of node | 1 |
| zone\_id | Indication of node’s physical location | 1 |

**File 1b: link.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| name | Optional for visualization purposes | Main Street |
| link\_id | Link identification number of the road | 1003 |
| from\_node\_id | Upstream node number of the link, must already be defined in *node.csv* | 1 |
| to\_node\_id | Downstream node number of the link, must already be defined in *node.csv* | 3 |
| link\_type\_name | Optional text label for visualization and data checking purposes | 1 |
| dir\_flag | Indication of directions of the link (=0, bi-direction; =1, single direction) | 1 |
| Length | The length of the link (between end nodes), measured in units of meter | 10 |
| free\_speed | Free-flow speed on defined link. Suggested Unit: mph | 60 |
| lanes | The number of lanes on the link | 1 |
| capacity | The number of vehicles per hour per lane (lane’s capacity per hour) | 2000 |
| link\_type | Index of link type name | 1 |
| toll | Optional generalized toll cost of the link, which could also be the cost of fuel | 0 |
| VDF\_fftt1 | Free flow travel time used in the volume-delay function (demand period in settings.csv) | 0.122391346 |
| VDF\_cap1 | Capacity used in the volume-delay function (demand period in settings.csv), VDF\_cap = demand period \* capacity \* lanes | 4000 |
| VDF\_alpha1 | Coefficient used in the volume-delay function | 0.15 |
| VDF\_beta1 | Coefficient used in the volume-delay function | 4 |

## 4.2 Input for demand data

* The specific files for demand layer are *demand.csv*
* Travel demand is given by periods. Thus, one file defines total volume of demand and one file defines time periods.

**File 2: demand.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| o\_zone\_id | Origin zone number of the link, must already defined in *node.csv* | 1 |
| d\_zone\_id | Destination zone number of the link, must already defined in *node.csv* | 2 |
| volume | Travel demand | 1500 |

## 4.3 Assignment and simulation configuration file

* The specific file for the configuration file is *settings.csv.*
* It defines basic setting for the Network, and contains five sections, namely assignment, agent\_type, link\_type, demand\_period, demand\_file\_list.

**File 3: settings.csv**

|  |  |  |  |
| --- | --- | --- | --- |
| **Section Name** | **Field Name** | **Description** | **Sample Values** |
| assignment | number\_of\_iterations | number of iterations for column generation | 20 |
| route\_output | produce route\_assgnment.csv or not (a binary value) | 1 |
|  | simulation\_output | Run DTA simulator or not (a binary value) to produce trajectory.csv and agent.csv | 0 |
| agent\_type | agent\_type | Abbreviation of the agent type | p; rt; dms |
| name | Full name of the agent type | Passengers or vehicles |
| flow\_type | Abbreviation of the flow type | 0 |
| vot | Value of time for the agent | 10 |
| Pce | Passenger Car Equivalent (PCE) of the agent | 1 |
| real\_time\_info | Real time information  0: no real time information, they will carry the original path from the ODME and early CG results. That is, no route change after the SA scenario is applied.  1: real time information allow any route change after the SA scenario is applied.  2: VMS: allow route change for the VMS zones to follow the real time path  Users have to define the zone\_id for each VMS node, we have a route flow merge process, first stage we use the original route, the second stage we switch to Real Time routes. | 0;1;2 |
| link\_type | link\_type | Index of link type name | 1 |
| link\_type\_name | The name of link type | motorway |
| demand\_period | demand\_period\_id | Demand period identification number (type: integer) | 1 |
| demand\_period | Name of the demand period (type: string) | AM |
| time\_period | Time period expressed by HHMM\_HHMM | 0600\_1200 |
| demand\_file\_list | file\_sequence\_no | Sequence number of reading the files | 1 |
| file\_name | Name of the file to be read | demand |
| format\_type | column for three columns, agent\_csv, routing policy | Column |
| demand\_period | Name of the demand period | AM |
| agent\_type | Abbreviation of the agent type | p |
| scale\_factor | Scale factor of different agent types (Sum to 1) | 0.2 |

**It should be noted that in the new version of DTALite**:

1. The settings.csv requires the users to enter the number of two iterations: **1) column generation iterations; 2) simulation iterations.** The users can decide whether to perform the simulation and its number of iterations.
2. In the original settings.csv file, there is the number of iterations of ODME, but in the updated version, **the code will determine whether there is a measurement file, if there is, the default number of iterations is 40**, and there are two conditions for stopping the iteration, one is to have iterated enough times, and the other is to have iterated to get a solution that cannot become better.

## 4.4 Input for supply-side scenario data

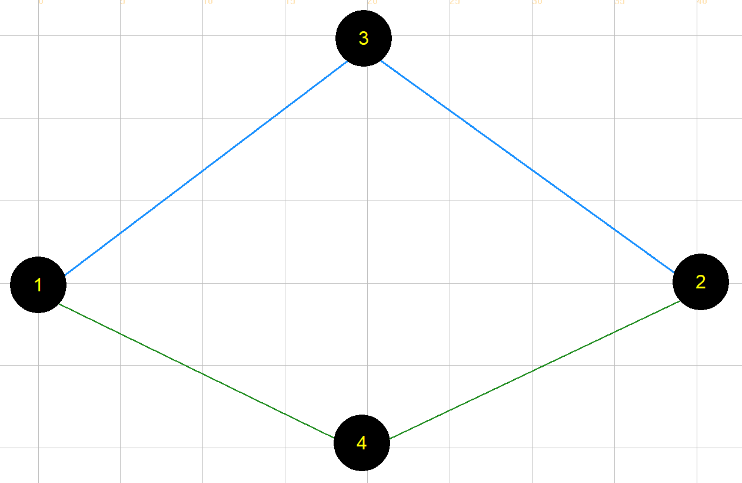
* The specific files for supply information are *supply\_side\_scenario.csv*
* Travel demand is given by periods. Thus, one file defines total volume of demand and one file defines time periods.

**File 4: *supply\_side\_scenario*.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| scenario\_type | Specific scenario defined in *supply\_side\_scenario.csv* | sa / rt / dms |
| from\_node\_id | Upstream node number of the link, must already be defined in *node.csv* | 1 |
| to\_node\_id | Downstream node number of the link, must already be defined in *node.csv* | 3 |
| lanes | Number of lanes after changing | 2 |
| demand\_period | Name of the demand period | AM |

See more details at [DLSim-MRM/supply\_side\_scenario.csv at main · asu-trans-ai-lab/DLSim-MRM (github.com)](https://github.com/asu-trans-ai-lab/DLSim-MRM/blob/main/dataset/4_101_corridor/supply_side_scenario.csv)

Let's use a small example (two-corridors network) to show how changing some condition values in the supply\_side\_scenario.csv file affects the results. The following figure shows a visualization of the example in NEXTA.



There are two paths in this figure:

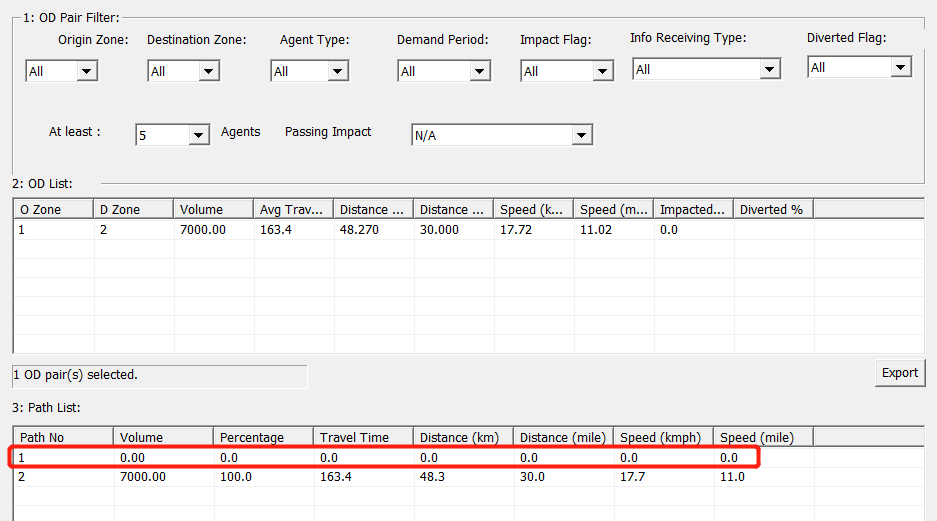
Path1: 1→3→2

Path2: 1→4→2

Here is the supply\_side\_scenario for the two-corridors network:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| scenario\_type | from\_node\_id | to\_node\_id | lanes | demand\_period |
| sa | 3 | 2 | 0 | AM |

See assignment in the NEXTA, the results are as follows:



We can see all agents choose the path 2, since path 1 (1→3→2) are not normal.

## 4.5 Output file

**File 5a: link\_performance.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| link\_id | Link identification number of the road | 1 |
| from\_node\_id | Upstream node number of the link, must already defined in input\_node.csv | 1 |
| to\_node\_id | Downstream node number of the link, must already defined in input\_node.csv | 3 |
| time\_period | The simulation period of the agent HHMM format | 0700\_0800 |
| volume | Link based flow volume for the defined period (volume per hour \* hour in the defined period) | 5600 |
| travel\_time | Link travel\_time in minute | 15 |
| speed | Average travel speed on the link | 38 |
| VOC | Volume /capacity ratio | 0.4 |
| notes | Some explanatory text | period-based |
| volume\_before | The volume of the link when the scene in the *supply\_side\_scenario.csv* is not executed | 6791.691 |
| volume\_after | The volume of the link after the scene in the *supply\_side\_scenario.csv* is executed | 6792.691 |
| volume\_diff | volume\_diff = volume\_after - volume\_before | 0 |
| speed\_before | The speed of the link when the scene in the *supply\_side\_scenario.csv* is not executed | 59.997 |
| speed\_after | The speed of the link after the scene in the *supply\_side\_scenario.csv* is executed | 60.997 |
| speed\_diff | speed\_diff = speed\_after - speed\_before | 1 |

**File 5b: route\_assignment.csv**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Sample Values** |
| path\_no | identification number of the path | 1 |
| o\_zone\_id | Upstream number of the zone, must already defined in zone.csv | 1 |
| d\_zone\_id | Downstream number of the zone, must already defined in zone.csv | 3 |
| od\_pair |  | 0 |
| o\_sindex |  | 1 |
| d\_sindex |  | 76->120 |
| within\_OD\_path\_no |  | 1 |
| path\_id | Path identification number of the route | 1 |
| agent\_type | Type of the agent | p, rt, dms |
| demand\_period | Name of the demand period | AM |
| volume | Path based flow volume for the defined period (volume per hour \* hour in the defined period) | 135.198 |
| simu\_volume |  | 0 |
| subarea\_flag |  | 1 |
| OD\_impact\_flag |  | 0 |
| at\_OD\_impact\_flag |  | 0 |
| vehicle\_diverted\_flag |  | -1 |
| toll | The amount of money/time that agent pays, unit: dollar | 0 |
| travel\_time | path travel\_time in minute | 10.3 |
| VDF\_travel\_time |  | 10.3251 |
| SA\_TT\_# |  | 10.328943 |

**File 5c: 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 | 2 |
| path\_id | Path identification number | 0 |
| o\_node\_id | Origin node number of the agent | 1 |
| d\_node\_id | Destination node number of the agent | 2 |
| agent\_type | Type of the agent | p |
| demand\_period | Name of the demand period | AM |
| volume | Flow volume assigned on the agent | 5600 |
| toll | The amount of money/time that agent pays, unit: dollar | 360 |
| travel\_time | The total time from the origin to the destination of the agent | 31.5 |
| distance | The total travel distance from the origin to the destination of the agent, unit: mile or km as users defined for length in link.csv | 20 |
| node\_sequence | The number of nodes through which agents pass in turn | 1;3;2; |
| link\_sequence | The number of links through which agents pass in turn | 1003;3002; |
| time\_sequence | Time point through which agents pass in turn | 0730:00;0745:45;0801:31; |
| time\_decimal\_sequence | The number of decimal times through which agents pass in turn | 450.00;465.76;481.52; |

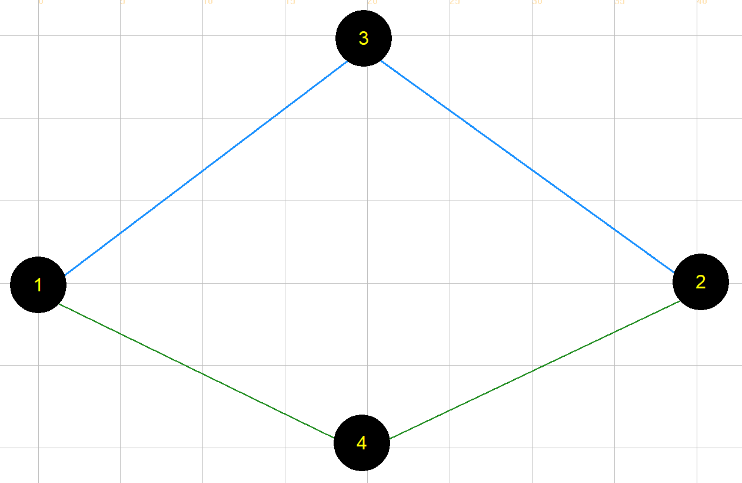
**File 5d: final\_summary.csv**

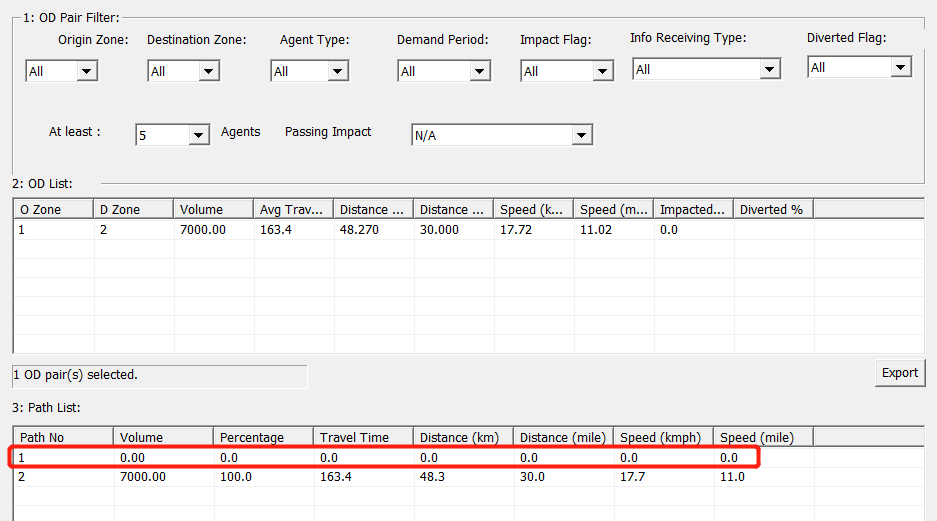
This file is a summary file of the output results, showing the summary of the final statistics.

|  |  |  |
| --- | --- | --- |
|  | **Steps** | **Information** |
| 1 | Read network node.csv | link.csv, zone.csv, summary by multi-modal and demand types, summary by road link type |
| 2 | Read demand | defined in [demand\_file\_list] in settings.csv. |
| 3 | Check OD connectivity and accessibility in od\_accessibility.csv | # of connected OD pairs, # of OD/agent\_type/demand\_type columns without paths, CPU Running Time, # of agents, Avg Travel Time, Avg UE gap |
| 4 | Column Generation for Traffic Assignment | Iteration, avg travel time, optimization obj, Relative\_gap |
| 5 | Column pool-based flow updating for traffic assignment | # of flow updating iterations，ODME stage，link MAE，link\_MAPE，system\_MPE，avg\_tt，UE gap |
| 6 | OD estimation | # of ODME\_iterations |
| 7 | Sensitivity analysis stage | Iteration，Avg Travel Time(min) |
| 8 | Column updating | Iteration，avg travel time，optimization obj，Relative\_gap |
| 9 | Output Link Performance | VMT，VHT，MPH，PMT，PHT，MAPE，VKT，PKT，KPH，simple avg link volume，simple avg link speed，simple avg link speed ratio，# of simulated agents in trajectory.csv |

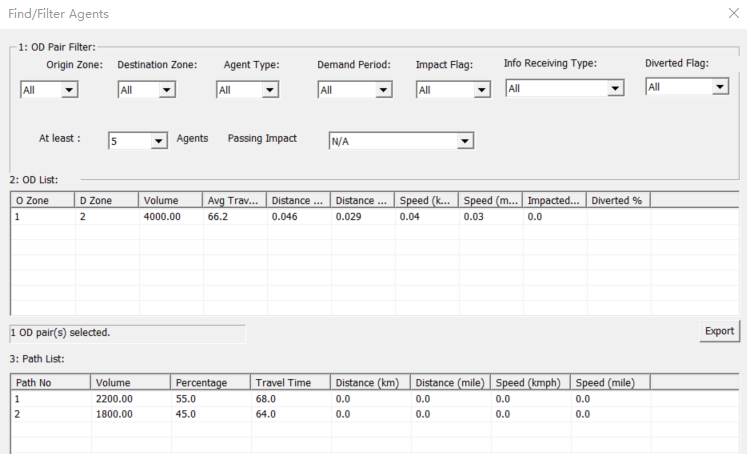
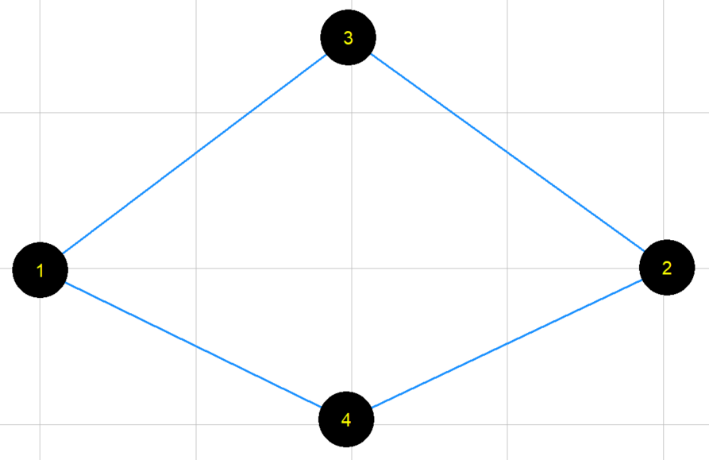
**Next, the results of route assignment will be shown with three datasets as an example.**

（1）01\_two\_corridor

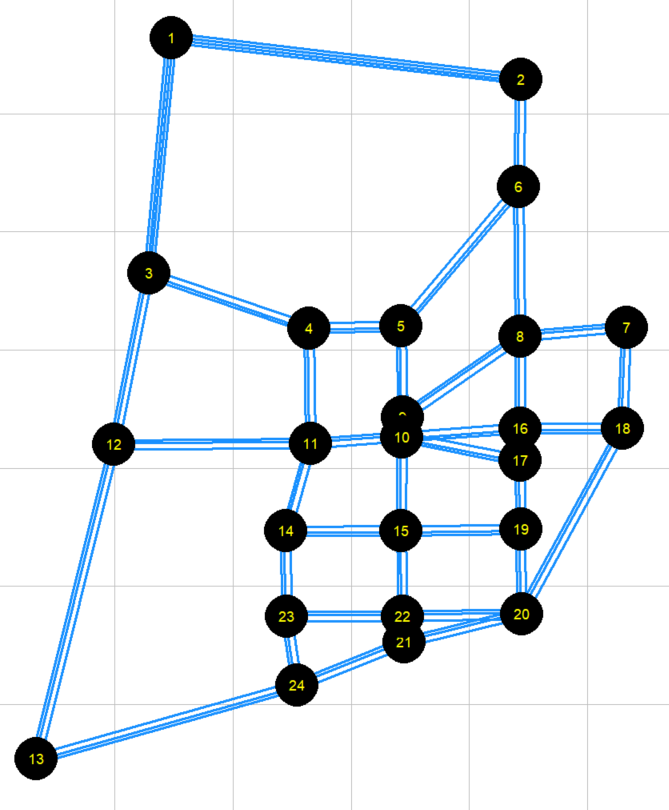


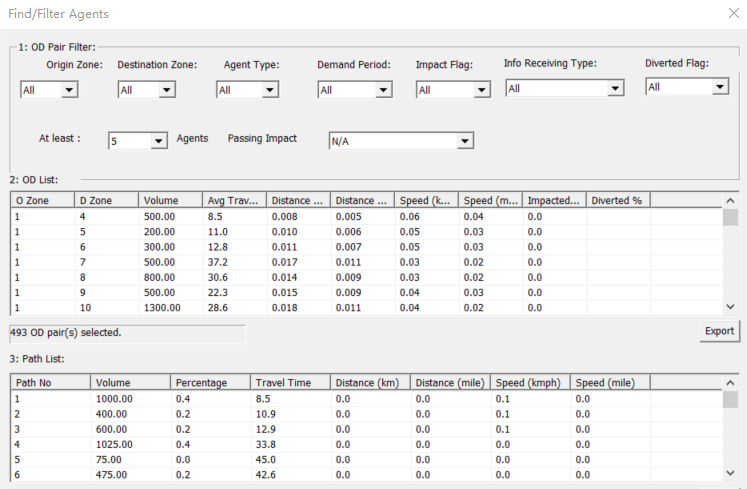


（2）02\_Braess\_Paradox

****

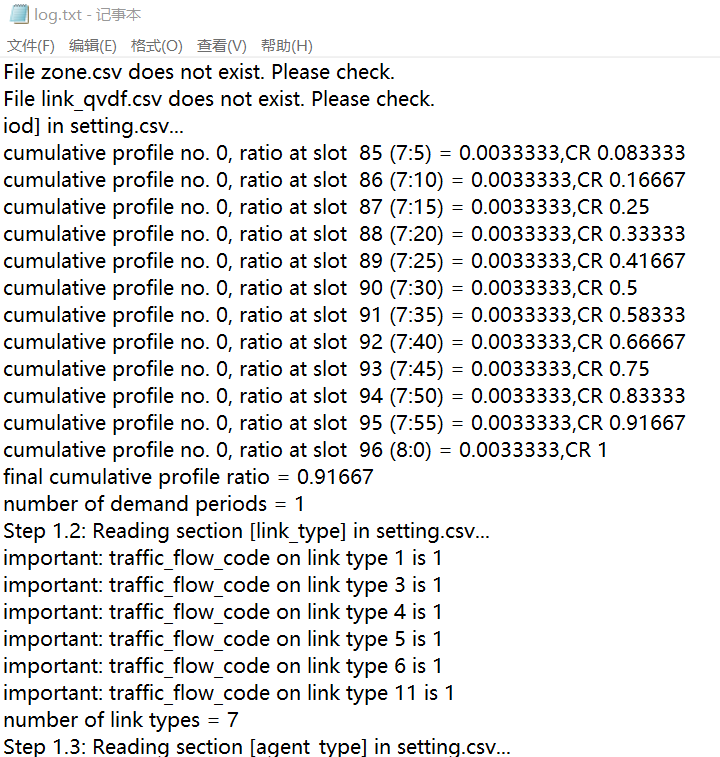
（3）04\_Sioux\_Falls\_network

****

****

**File 5e: log.txt**

This file is the file that records the results of the program running process, some parameters and results generated during the DTALite running process can be found in this file.



**It should be noted that in the new version of DTALite**:

1. In the **route\_assignment.csv** file there are **three types of od: 1) p; 2) rt; 3) dms.** What we need to pay attention to is the travel time corresponding to the three types of od, check whether the travel time of rt have better travel time than p and dms (rt should have better performance than p and dms).
2. The value of the **od\_impact\_flag** can be 0 or 1, when it is 1, it means that the current od pair is the one that will be affected when there is a traffic accident.

## 4.6 Introduction to the Datasets

Each dataset has its own special application scenarios and test implications, you can download the datasets from [DTALite/dataset at main · asu-trans-ai-lab/DTALite (github.com)](https://github.com/asu-trans-ai-lab/DTALite/tree/main/dataset).

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Targeted scenario | Key output files | Key output performance |
| 01\_two\_corridor | UE | Route\_assignment.csv | Two routes have equal travel time |
| 02\_Braess\_Paradox | Network design: add link, delete link | Link.csv | Total travel time; link performance before and after changing |
| 03\_three\_corridor | Demand changes | UE: route\_assignment | Two routes have equal travel time; information of route |
| 04\_Sioux\_Falls\_network | Real world case （24 nodes） | Standard link\_volume \_performance | link performance: volume, speed, congestion duration |
| 05\_Chicago\_Sketch | Real world case（933 nodes） | Standard link\_performance | link performance:  volume, speed, congestion duration |

# Appendix A: From mathematical modeling to network-based assignment and simulation

1. **Link volume-delay function in static traffic assignment**

There are a number of key components for the static traffic assignment procedure.

* input trip table describes the flow per hour from each origin zone to each destination zone
* a traffic network consisting of nodes, links and link volume delay functions
* volume-delay function such as BPR **(Bureau of Public Roads** **)** relationship that shows increased link travel time as an increase in link volume

TT = FFTT[1 + 0.15(v/c)4]

where:

TT = link travel time

FFTT= free-flow travel time of link

v = link flow

c = link capacity

[Remark: the link travel time function typically is only dependent on its own flow, while ignoring link volume on opposing or conflicting directions. The link capacity might not be a strict upper limit on flow, e.g. specified by highway capacity manual.]

As one of thesimplest*c*ases of behavior, User Equilibrium (UE) Principle assumes users are“greedy” and are familiar with the system. E*quilibrium requires iteration* to reach the following two principles:

* Principle A: No individual trip maker can reduce his path costs by switching routes.
* Principle B: All used routes between an O-D pair have equal and minimum costs

While all unused routes have greater or equal costs *(to the used path costs)*.

Wardrop (1952) proposed the user equilibrium and system optimal principles of route choice behavior in his seminal paper, and Beckman et al. (1956) formulated the static user equilibrium traffic assignment problem as an equivalent convex mathematical programming problem. Since their influential contributions, the development of the static network assignment formulations, algorithms and applications have made remarkable progress. The books by Sheffi (1985) and Patriksson (1994) provide the most comprehensive coverage on the static traffic assignment problem and its variants.

1. **General mathematical descriptions of traffic assignment**

Traffic assignment loads an origin-destination (OD) trip matrix onto links of a traffic network, while satisfying a certain route choice behavioral model, e.g., deterministic user equilibrium. Traffic assignment is used to predict/estimate how trip-makers may shift to other routes or departure times in response to a number of strategies such as road pricing, incidents, road capacity improvement and traffic signal re-timing.

For example, tolling typically lead to traffic diversion on alternative routes and/or other transportation modes, and many traffic congestion mitigation strategies should be developed to improve the capacity to which the traffic may be diverted, for example, signal optimization, traveler information provision, and transit operation.

The common time periods include morning peak, afternoon peak and off-peak, and we can use the time of day factor to calculate the trip in the peak hour (e.g., morning peak may be 11% of daily traffic) from a 24 hour demand volume.

By using a simplified static traffic assignment formulation, the following mathematic description adopts the related sections in the paper titled “Equivalent Gap Function-Based Reformulation and Solution Algorithm for the Dynamic User Equilibrium Problem” by Lu, Mahmassani and Zhou in (2009). One can consider the extended DTA formulation by adding a time index dimension.

Consider a network G = (*N*, *A*), where *N* is a finite set of nodes and *A* is a finite set of directed links (*i*, *j*), *i*∈*N* and *j*∈*N*. Associated with each link (*i*, *j*) is the link travel time *sij*(*t*) required to traverse link (*i*, *j*) when departing at time interval *t*∈*S* from node *i*. For simplicity and without loss of generality, *sij*(*t*) is regarded as link travel time, though it can be generalized to include travel time, out-of-pocket cost and other travel impedances that may incur when traversing link (*i*, *j*) at time *t*. Travel time and cost are used interchangeably in this paper. Other important notation and variables are summarized below.

*O* subset of origin nodes; *O ⊆ N*

*D* subset of destination nodes; *D ⊆ N*.

*T* set of departure time intervals.

*o* subscript for an origin node, *o*∈*O*.

*d* subscript for a destination node, *d*∈*D*.

set of all feasible paths for a given triplet (*o*, *d*).

*p*  subscript for a path *p*∈.

number of trips departing from node *o* to node *d*.

number of trips departing from *o* to *d* and assigned to path *p*∈.

*r* path flow vector, *r* = {, ∀*o* ∈*O*, *d* ∈*D*, and *p*∈ }.

path travel cost (or time) for the travelers departing from *o* to *d* and assigned to path *p*∈;, and is a function of the path flow vector *r*.

*c*(*r*) vector of path travel costs; *c*(*r*) = {, ∀*o* ∈*O*, *d* ∈*D*, and *p*∈ }.

The OD demand pattern for the entire planning horizon (i.e.,, ∀*o*, *d* is assumed to be known *a priori*. The key behavioral assumption for the path choice decision is as follows: in a disutility-minimization framework, each trip-maker is rational and chooses a path that minimizes the travel cost. Specifically, for each trip-maker in, a path *p*\*∈ will be selected if and only if .

Given the assumptions above, the problem is to solve the UE traffic assignment problem, with a given OD demand, to obtain a path flow pattern satisfying the UE conditions. Specifically, the goal is to determine a UE path flow vector (routing policies) over a vehicular network for each OD pair and each departure time interval (i.e., *r\** ≡ {, ∀*o*, *d*, and *p*∈ }.

By the above UE definition, all trips in a network are equilibrated in terms of actual experienced path costs, so it is necessary to determine the experienced path costs *c*(*r*) for a given path flow vector *r*. To this end, a simulation-based dynamic traffic (or network loading) model is used to obtain the experienced path cost vector. It should be noted that the algorithm is independent of the specific dynamic traffic model selected; any (macroscopic, microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics, in particular the effect of physical queuing, as well as preventing violations of the first-in-first-out property, can be embedded into the proposed solution algorithm.

With the introduction of the gap function *Gap*(*r*, *π*), the proposed nonlinear minimization problem (NMP) is presented as the following.

(1)

Subject to , ∀*o*, *d* (2)

, ∀*o*, *d*, and *p*∈*P*(*o*, *d*) (3)

, ∀ *o*, *d*, and *p*∈*P*(*o*, *d*) (4)

In the above NMP reformulation, both *π* and *r* are *independent* decision variables and hence the gap function is a function of both *r* and *π* (i.e., *Gap*(*r*, *π*)), where *π* and *r* are connected with each other through inequality constraint (3). *Gap*(*r*, *π*) provides a measure of the violation of the UE conditions in terms of the difference between the total actual experienced path travel cost and the total shortest path cost evaluated at any given path flow pattern *r*∈Ω. The difference vanishes when the path flow vector *r*\* satisfies the UE conditions. Thus, solving the UE problem can be viewed as a process of finding the path flow vector *r*\*∈Ω and *π*\* such that *Gap*(*r\**, *π*\*) = 0.

[DTALite/dataset at main · asu-trans-ai-lab/DTALite (github.com)](https://github.com/asu-trans-ai-lab/DTALite/blob/main/dataset/02_Braess_Paradox/Braess_network_Process_Tutorial.xlsx)

# Appendix B: The method of preparing the zone ids

Three files are required for processing dynamic traffic assignments, TAZ.csv, node.csv, and link.csv. In the classic version of DTALite, the zone-to-node mapping is specified in node.csv through the field of zone\_id, as Figure 1 shown. However, there is a challenge when mapping TAZ-based information to the network at different resolutions.

Graphical user interface, application, table, Excel

Description automatically generated

**Figure 1** node.csv

Thus, in the preprocessing stage, we allow users to prepare (1) TAZ with boundary information, and then (2) DTALite will generate the zone.csv as an explicit mapping between zone\_id and access node vector. Then users can still manually edit the accesss\_node\_vector to refine the zone-to-node mapping in zone.csv

In the assignment stage, DTALite reads the zone-to-node mapping from zone.csv or node.csv, and reads the related OD demand table (defined in terms of o zone id and d zone id) to assign the OD volume to different routes connecting the origin activity node to the destination activity node.

This guide provides an instruction to help users prepare the TAZ.csv (take Phoenix as an example).

**Step 1** Download census shapefiles of Phoenix on the [DATA.GOV](https://catalog.data.gov/dataset/tiger-line-shapefile-2011-2010-state-arizona-2010-census-traffic-analysis-zone-taz-state-based-)

In addition, we summarize the other two websites to download shapefiles: 1) we can use census tracts as TAZs, the shapefiles of census tracts for all US states are available [here](https://www.census.gov/cgi-bin/geo/shapefiles/index.php?year=2021&layergroup=Census+Tracts) and 2) for some cities can be found [here](https://koordinates.com/layer/96425-phoenix-arizona-census-tracts-2010/).

1. Search the Census data.

Graphical user interface, text, website

Description automatically generated

2. Download one dataset which includes a .shp file.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Step 2** Use QGIS to export TAZ shapefile as TAZ.csv

1. Right-click the TAZ layer -- "Export"-- "Save Features As"

A picture containing map

Description automatically generated

2. Save the features as TAZ.csv

Set up the Format as CSV, and the CRS as WGS 84;

Select fields of "TAZCE10", "INTPTLAT10" and "INTPTLON10" to export;

Set up the Geometry type as Polygon;

Select the format of GEOMETRY;

OK

Graphical user interface

Description automatically generated

3. Change the fields' name of "WKT", "TAZCE10", "INTPTLAT10" and "INTPTLON10" as " geometry", "TAZ", "y\_coord", and "x\_coord " repectively.

Text

Description automatically generated Graphical user interface, text

Description automatically generated

Note: we need to use a text editor to open the TAZ.csv, since there could be some display errors if we use Excel to open it due to the limitation of string length.

References:

[jdlph/Path4GMNS: An open-source, cross-platform, lightweight, and fast Python path engine for networks encoded in GMNS. (github.com)](https://github.com/jdlph/Path4GMNS)

[jdlph/shortest-path-algorithms: Boost Shortest Path Algorithm Implementations using Proper Data Structures (github.com)](https://github.com/jdlph/shortest-path-algorithms)

[Transportation Network Analysis -- Boyles, Lownes, and Unnikrishnan (sboyles.github.io)](https://sboyles.github.io/blubook.html)

Calendar

Description automatically generated with medium confidence

Parallel computijng for using different processors

Use nexta to create subarea.csv

Menu ->

Graphical user interface, chart, application

Description automatically generated

En sure Route\_output = 1 in settings.csv

Check subarea\_related\_zone.csv

There are zone\_id and super\_zone\_id

Super\_zone\_id = -1 means this zone is not considered in the final assignment.

total pre-load demand = 1.93976e+06

1344 zones are not significantly related by the subarea. = (0.34828)

243 zones are inside the subarea.

total demand = 1.979e+06 total subarea-related demand = 2.7223e+05 (0.13756 )

if we load route.csv we will ignore the column generation and ODME process.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | route.csv based implementation | | |
| subarea.csv | |  |  |  |  |  |  |  |  |  |  |
|  | focusing approach |  |  |  |  |  |  |  | skip |  |  |
|  |  | preload demand |  |  |  |  |  |  |  |  |  |
|  |  | related OD pairs passing through this area |  |  |  |  |  |  |  |  |  |
|  |  | affected zones |  |  |  |  |  |  |  |  |  |
|  |  | zone id to super zone id mapping |  |  |  |  |  |  |  |  |  |
|  |  |  | subarea\_related\_zone.csv | | |  |  |  |  | read subarea\_related\_zone.csv | |
|  |  | clustering | all zones in the subareas are kept as super\_zone | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | column generation |  |  |  |  |  |  |  | read route.csv | |
|  |  | column updating |  |  |  |  |  |  |  |  |  |
| measurement.csv | | ODME |  |  |  |  |  |  |  |  |  |
| supply\_side\_scenario | | column\_regeneration |  |  |  |  |  |  |  |  |  |
|  |  | SA: sensivitity analysis -> UE | hist |  |  |  |  |  | apply SA |  |  |
|  |  |  | real time |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | VMS | VMD nodes-> zone\_id, must be also super\_zone\_id | | | | |  |  |  |
| Final results | |  |  |  |  |  |  |  |  |  |  |
|  |  | link\_performance.csv |  |  |  |  |  |  |  |  |  |
|  |  | subarea\_link\_performance.csv |  |  |  |  |  |  |  |  |  |
|  |  | route\_assignment.csv | > the base of route.csv | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |