**User Guide of Big data-driven Transportation Computational Graph**

Please send your comments to [xzhou74@asu.edu](mailto:xzhou74@asu.edu) if you have any suggestions and questions.

This document serves as a simple user guide for the GMNS data format version of Big data-driven Transportation Computational Graph (BTCG) framework.

1. **Introduction**

BTCG (Wu et al., 2018) is a forward and backward propagation algorithmic framework on a layered computational graph, which can perform hierarchical travel demand estimation using multiple data sources. BTCG can be viewed as an implementation of conceptual Hierarchical Flow Networks (HFN). By applying the back propagation (BP) algorithm, one can view each variable as a vertex and the edge between vertexes to translate the calculation process between variables as a computational graph.

The specific relationship between the layers of HFN is shown in Fig 1., which covers the input/output variables of each layer and the relationship between layers.

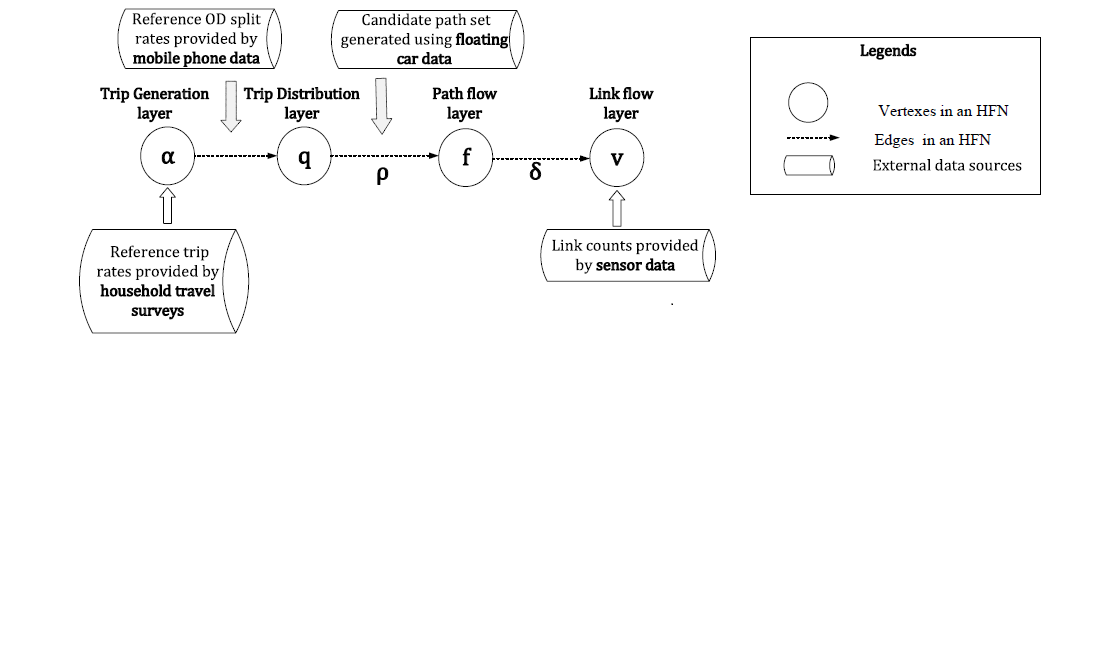


Fig. 1. HFN of Traffic Demand Flow Estimation (TDFE) model

To minimize the loss function, parameters such as α, π and θ are updated during the training process. Finally, the traffic flow of each layer (e.g., α of ozone layer, γ of OD layer, v of link layer) are estimated jointly.

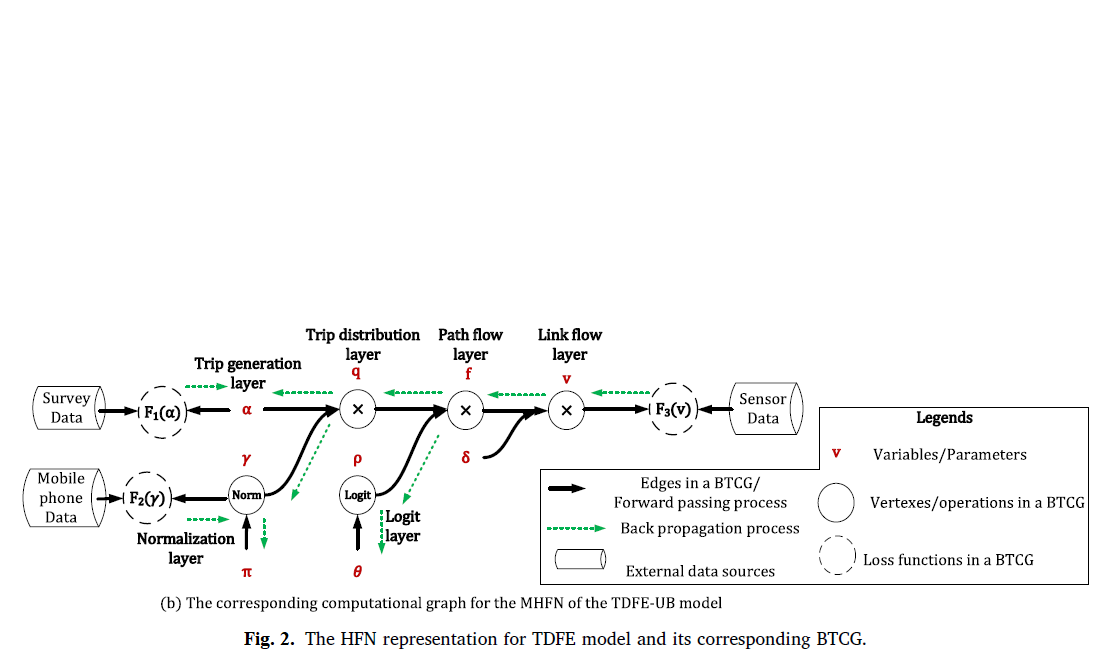


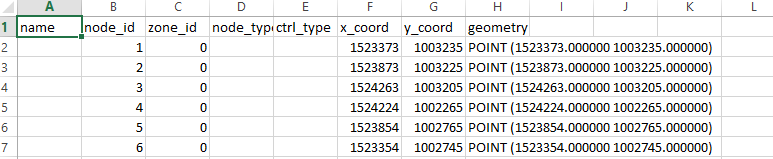
Fig. 2. Corresponding BTCG of HFN

1. **Data flow**

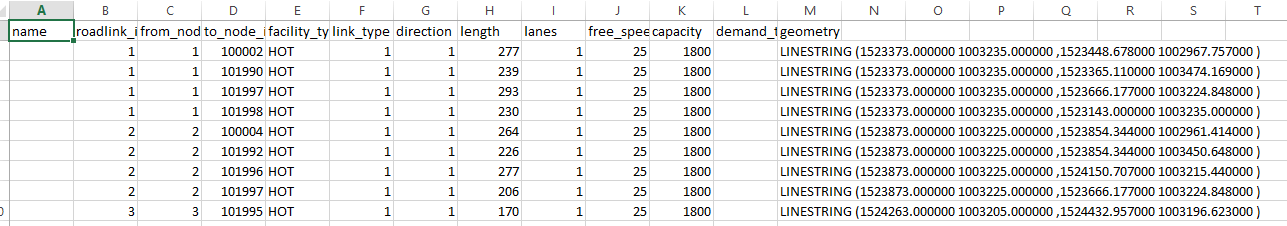
|  |  |
| --- | --- |
| **Input files** | **Output files** |
| node.csv | output\_ozone.csv  output\_od.csv  output\_path.csv  output\_link.csv |
| agent.csv  agent\_type.csv |
| road\_link.csv |

File node.csv

This file of node.csv includes the basic node information about the test network, such as name, node\_id, zone\_id, node\_type, ctrl\_type, x\_coord, y\_coord, and geometry.

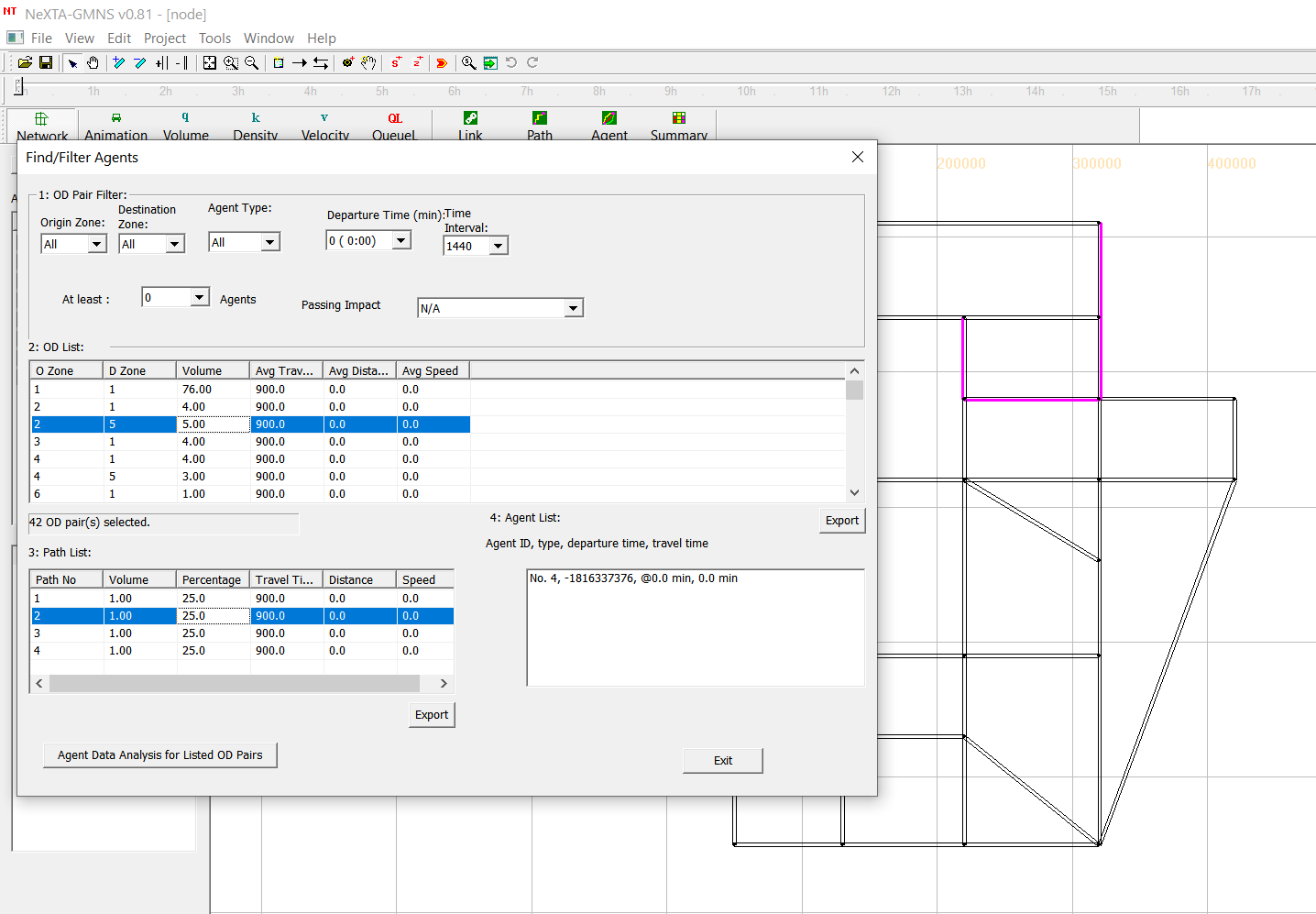


road\_link.csv, which includes basic link-level information.

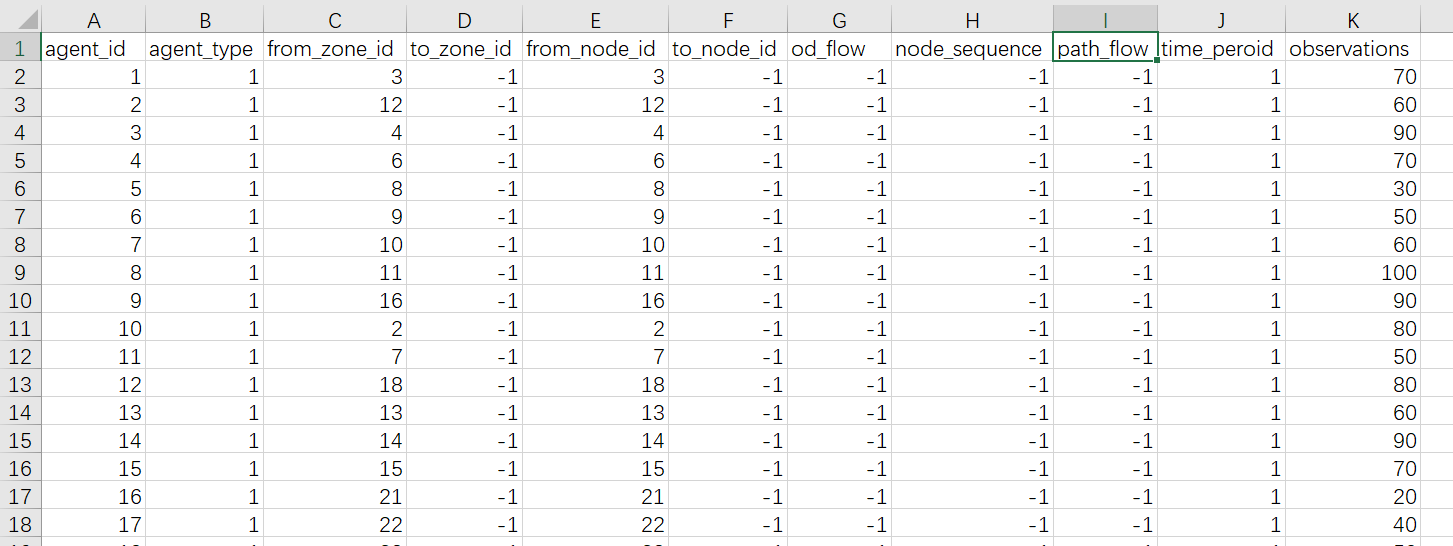


input\_agent.csv

There are different types of measurements stored in agent.csv, and one can use NeXTA to visualize the path trajectories.

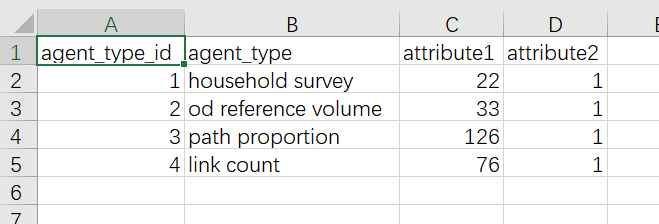


Our source code considers an integration of multiple data sources, namely the household survey data (ozone data), the OD reference volume or the OD split rate (mobile phone data), the link count (sensor data), and the path information such as its node sequences.



agent\_type.csv

File agent\_type.csv is used to specify the agent/information type in the measurements in input\_agent.csv. Currently, we consider 4 information types: 1. household survey samples; 2. OD reference volume; 3. path proportions; 4. link counts. The field of “attribute 1” means the number of observations, and the field of “attribute 2” means the number of time periods. In our current implementation, we consider one day or peak period of traffic volume, thus the number of time periods is set to 1.



1. **Case study**

As shown in Fig. 3, the illustrative Sioux Falls network is used as the test case, with 22 zones, 24 nodes, 33 OD pairs, 76 links and 132 paths. The GMNS format data set is provided in the sub folder of /SiouxFalls network.

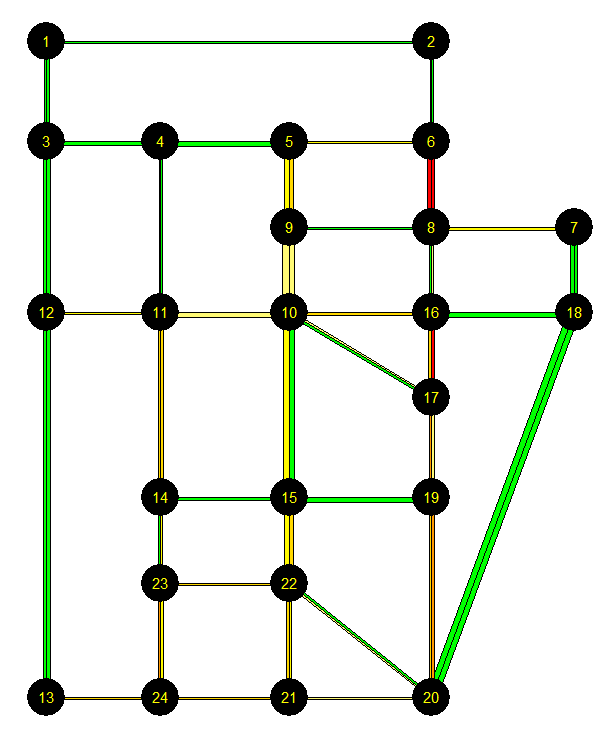
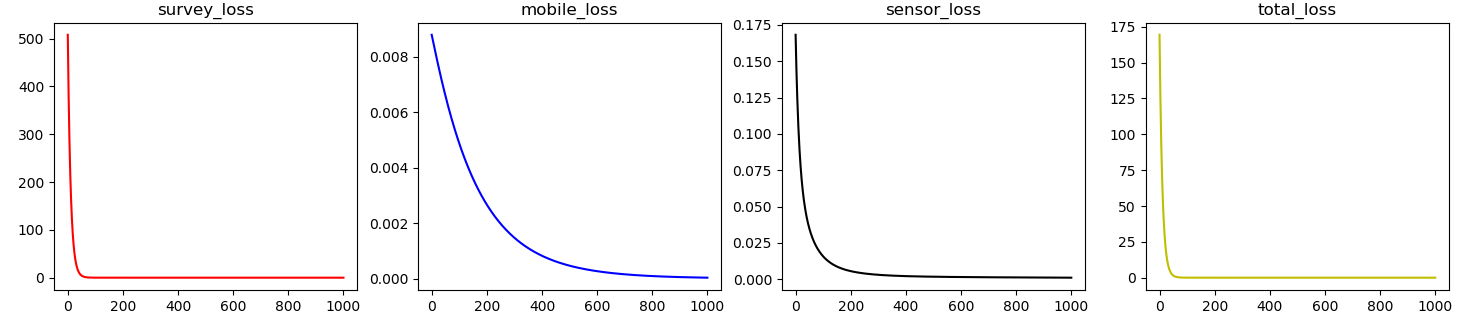


Fig. 3. Sioux Falls Network

Training results (1000 epochs)：

BTCGLite\_GMNS.py. Tthis simplified version does not use the logit model and related class objects, to speed up the computational time.



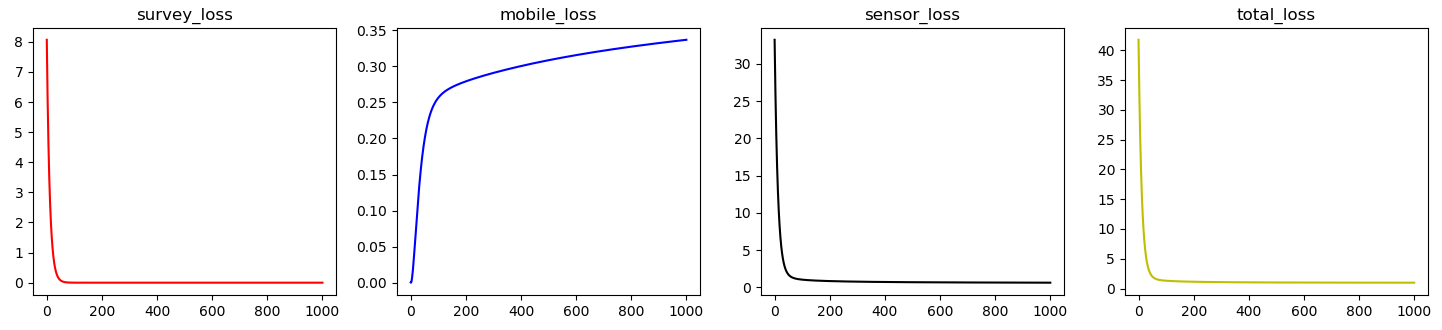
step 1000 :survey error= 6.983434152935154e-08

step 1000 :mobile error= 3.8845606447143905e-05

step 1000 :sensor error= 0.0010404423796513053

step 1000 :total error= 0.0010792072388590332

BTCG\_GMNS.py In this standard version with embedded non-convex logit functgions, the loss associated with mobile data (i.e. OD splits) increases to a certain degree, but the total loss function has been reduced as a result ot the optimization process performed by Tensorflow.



step 1000 :survey error= 3.1824046e-09

step 1000 :mobile error= 0.3365797

step 1000 :sensor error= 0.6412415

step 1000 :total error= 0.9778478

**References.**

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transportation Research Part C: Emerging Technologies 96, 321-346.