IFN User Guide

Version 0.1.3

Kardi Teknomo

This document is the user guide to use the python code for Traffic Assignment based on Ideal Flow Network (IFN). The current version of this program should be run in Python 3.7 or higher version. The code consists of the following python files:

File Name	Description
`ifn.py`	The main program for traffic assignment
`display_network.py`	To display a network congestion level
`IdealFlowNetwork.py`	Ideal Flow Network library

You do not need to know python programming language to use the program. You need to download python 3.7 or above including Idle. If you know python programming language, it would be your advantage because you can analyze the results in more details than what explained in this document.

What is Traffic Assignment?

Almost everyone hates traffic congestion. What can we do about it? Almost everybody has some kind of ideas to solve the traffic congestion that we face every day. The problem is how to quantify our ideas such that we can evaluate which among our ideas would really work. This is where you need the tool, called traffic assignment software, to help you in modeling the road network, get the current network performance, and then create scenarios to alter the current road network on computer based on your ideas. Comparing the network performance of these scenarios would help you understand which ideas would work and which ideas would not work and which best ideas should be implemented.

Implementing your ideas directly in the actual road construction would cost a lot of time and money and the public would suffer in your experiment. Instead, you should try first all the possible ideas on computer and then you can propose to your government on your ideas. Knowing the exact amount of saving based on the propose scenario would also help you in quantifying the justification of the actual construction.

Given a road network, we would like to know what would be the flow and congestion level and network performances. Traffic assignment is a model to help us in assigning the flow and determine the congestion level on each link. Knowing the congestion level on each link, the program would also help us in finding the other link characteristics such as speed, travel time, delay and other network performances. Our goal in creating this program is to equip you with the best tools to test your creativity such that it can democratize the task to solve the traffic congestion in your own city. We are hoping that *anyone* (not necessarily as transportation engineer or transportation expert), equipped with these tools, can help in solving the traffic congestion quantitatively through science rather than merely based on opinions.

Many existing traffic assignment model exists but in general, most of them suffer several problems:

1. The traffic assignment software are **very expensive** (about $10,000 per license and you need to pay maintenance fee per year).

2. Many traffic assignment software are **very complicated**. The commercial demos are very nice with very lucrative animation but when it comes to solve the real world problem, you start to get a lot of doubt. You need to be a transport expert to input the data and to run the program.
3. Many traffic assignment software **requires extensive data**, especially Origin-Destination (OD) demand data. The data input are tremendously very expensive to gather and without those data, you are not able to model properly. If you input with any data, then garbage in garbage out.
4. The algorithm inside these commercial software are often heuristics and you treat them as black box without knowing how does it work and what are the assumptions behind the black box.

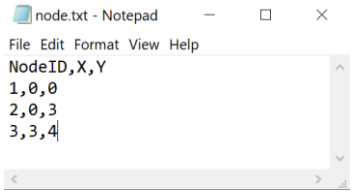
In contrast, IFN is open source, free to use and free to modify and free to distribute. The usage is relatively very simple, which will be explained in this user guide and it does not requires extensive data. The ideal flow network itself has strong mathematical background, based on Markov Chain and Maximum Entropy maximization whose assumptions are clearly stated. The IFN model is based on mathematical theory and not based on heuristic approach. The more data you have, it would be more accurate but at the most parsimony level, the model can be run without any data aside from the network itself. Based on the maximum entropy principle, we assume the maximum doubt when you have no data.

IFN Tutorial

IFN requires input of a network structure (nodes and links that connects the nodes). Let us prepare our first simplest network, which consist of three nodes and three links.

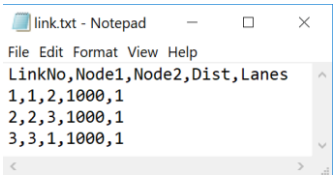
1. Create a text file and copy and paste the following, then save it as `node.txt`.

```
NodeID,X,Y
1,0,0
2,0,3
3,3,4
```



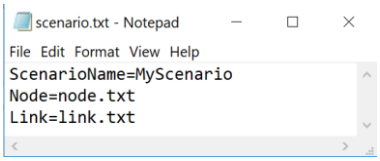
2. Create another text file and copy and paste the following, then save it as `link.txt`.

```
LinkNo,Node1,Node2,Dist,Lanes
1,1,2,1000,1
2,2,3,1000,1
3,3,1,1000,1
```



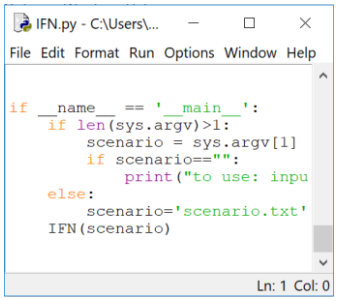
3. Create another text file and copy and paste the following, then save it as `scenario.txt`.

```
ScenarioName=MyScenario
Node=node.txt
Link=link.txt
```

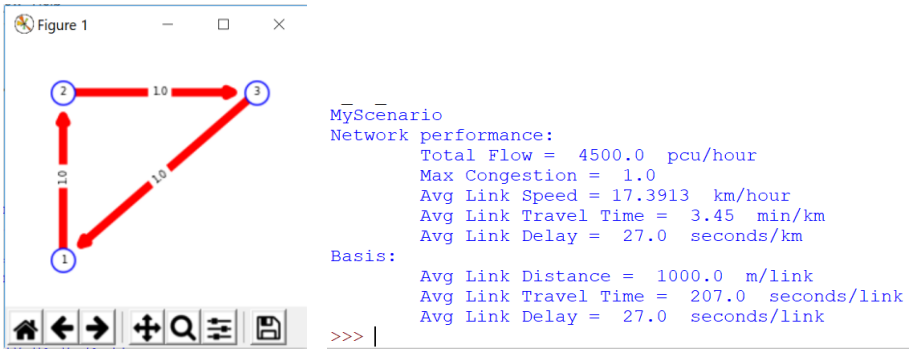


4. Open `ifn.py` using Idle (the official IDE of python) and at the bottom, change the scenario file into

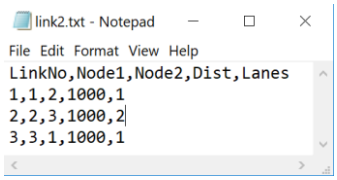
```
if __name__ == '__main__':
        …
        scenario='scenario.txt'
    IFN (scenario)
```



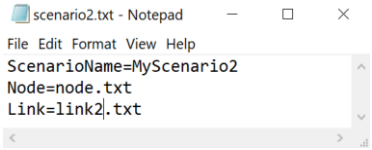
5. Run the core_ta_ifn.py and you get the result. The values in the links in the network represent the link congestion level.



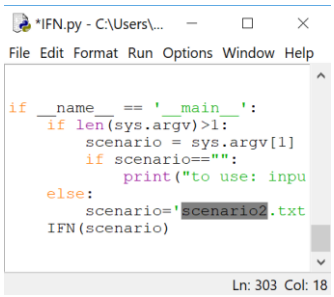
6. Now you can alter the link file into link2 where the second link from node 2 to node 3 would have 2 lanes. Save this link file as `link2.txt`.



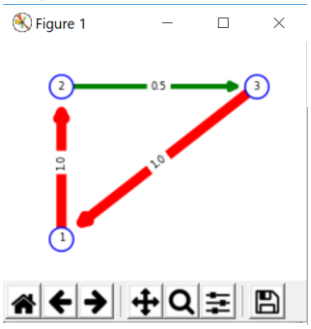
7. Now we need to modify the scenario where we still use the same nodes but only the number of lane is changing by changing the link file and the scenario name in the scenario file. Save the scenario file as `scenario2.txt`.



8. Modify the scenario name in `ifn.py` and then run the program



9. The results can be seen and compare with the previous scenario. The average speed is improving with the additional lane from 17 km/hour into 23 km/hour. The travel time is also improving from 3.45 min/km now becomes 2.88 min/km. Similar result happen with the delay.




```
MyScenario2
Network performance:
        Total Flow =  4500.0  pcu/hour
        Max Congestion =  1.0
        Avg Link Speed = 23.1525  km/hour
        Avg Link Travel Time =  2.8768  min/km
        Avg Link Delay =  18.3214  seconds/km
Basis:
        Avg Link Distance =  1000.0  m/link
        Avg Link Travel Time =  172.6071  seconds/link
        Avg Link Delay =  18.3214  seconds/link
```

Agreement of the Inputs

Each scenario in IFN requires three files: scenario file, node file and link file.

Scenario File

At the minimum, the scenario file consists of scenario name, a pointer to the node file and a pointer to the link file. The keywords on the left hand side of the equal sign must be kept unaltered. The right hand side of the equal sign can be anything.

```
ScenarioName=MyScenario
Node=node.txt
Link=link.txt
```

The scenario name would be used to determine the name of the output files.

Inside the scenario file, you can also add additional parameters of the model based on the following keywords. The keywords must be on the left hand side (LHS) of the equal sign. The order of the keyword does not matter.

LHS keyword	Optional RHS	Description
ScenarioName	Any scenario name	The name of output file in the same folder
Node	The file name of node file	Node file must be in CSV format
Link	The file name of link file	Link file must be in CSV format

maxAllowableCongestion	A float number (between 0 and 1)	Maximum allowable congestion parameter will be used if the calibration basis = congestion. If you do not specify, the default value is 1.
totalFlow	An integer number	Total flow parameter will be used if the calibration basis = flow
travelTimeModel	{`BPR`, `Greenshield`}	You can specify which between the two models of travel time would be used by the program. If you do not specify the default is BPR.
calibrationBasis	{`flow`, `congestion`}	Specify the basis for calibration. If the calibration basis = flow then it would maintain the total flow in the network (including dummy links). This is the same as maintaining the same demand. If the calibration basis = congestion then it would maintain the maximum congestion level to be equaled to specified maximum allowable congestion parameter. If you do not specify the calibration basis, the default value is set to congestion.
capacityBasis	{`lanes`, `width`}	The program would convert either the number of lanes or the road width in each link into a link capacity. If your input link data in column 5 is the number of lanes, then you need to specify capacity basis = lanes. If your input link data in column 5 represent the road width (in meter), then you need to specify capacity basis = width. If you do not specify the capacity basis, the program will assume the default value as the number of lanes.
cloudNode	Integer Node-ID of the cloud node	When you use a cloud node, you need to specify the node-ID of the cloud node.

Node File

A node file can be any name but the format must be in CSV. The first row is fixed header

```
NodeID,X,Y
```

The header itself can be in any language but the location of each field must be fixed. For instance, the following header is also valid.

```
myPointID,longitude,latitude
```

From the second row, you can specify the actual node-ID and the coordinate location in X and Y. The node-ID must be an integer number. There is no limit of the number of node and the node-ID is not necessarily in order. The coordinate location can be floating numbers or integers.

Link File

A link file can be any name but the format must be in CSV. The first row is fixed header

```
LinkNo,Node1,Node2,Dist,Lanes
```

The header itself can be in any language but the location of each field must be fixed. For example, the following header is also valid because the meaning of the field is unaltered.

```
Number,firstNode,secondNode,length,roadwidth
```

From the second row, you can specify the actual data for each link.

1. LinkNo is any link ID, preferable in integer. There is no limit of the number of links in the network. The link-ID is not necessarily in order.
2. Node1 is the first nodeID of the link. Since a link is a directed link, it requires two nodes in order.
3. Node2 is the second nodeID of the link. The arrow of the directed link is from node1 to node2.
4. Distance represents the length of each link, in meter.
5. Lanes represents the number of road lanes. The Lanes (column 5) can be replaced with Road width in meter. The default is Lanes and if your link data consist of road width instead of the number of lanes, then you need to specify **`capacityBasis=width`** in the scenario file.

The program will first automatically compute the maximum speed and the capacity of each link before computing everything else. That means if you have your own maximum speed and link capacity for all links, then you can specify the links file as follow:

```
LinkNo,Node1,Node2,Dist,Lanes,MaxSpeed,Capacity
```

In that case the Lanes (or Road-width in column 5) of the link data would be ignored and the program will use directly the maximum speed and capacity that you specify.

Calibration Basis

The IFN would automatically calibrate the result based on certain assumptions. You can select between the two assumptions below:

1. Maintain the maximum congestion level
2. Maintain the demand of flow in the entire network

Maintaining the maximum congestion level would require you to specify the maximum allowable congestion parameter. The default value of the maximum allowable congestion is one. This option would set the maximum congestion level to your specification. It means you want to keep the maximum congestion in any link to be a constant among the scenarios that you are comparing. This option will help you to know what would be the maximum demand of flow that you can accommodate in your network.

Maintaining the total flow in the network indicates that you want to keep the demand to be constants among the scenarios that you are comparing. This option will set the total flow (including the flow in the dummy links) to be constant as to you total flow specification. This option will help you to know what would be the maximum congestion level given the total flow demand in the network.

In the scenario file, you can optionally specify the basis of calibration. If you set **`calibrationBasis=flow`** then the program maintains the total flow in the network. This means you want to keep the total demand in the network to be invariant among all the scenarios. If the **`calibrationBasis=congestion`** then it would maintain the maximum congestion level to be equaled to specified maximum allowable congestion parameter. If you do not specify the calibration basis, the default value is set to congestion.

Travel Time Model

There are two options of travel time models being used in IFN:

1. Greenshield model
2. BPR model

Using Greenshield's traffic model, we assume the speed-density relationship is linear and the congestion level (which is equal to the flow/capacity) is set to be between zero and one. Since the congestion level is normalized to be between zero and one, it is easier to interpret the meaning of congestion level. However, the Greenshield tends to have higher speed than BPR (for the same congestion level) and only operates when the traffic is not so congested.

BPR model produce better variation of speed and travel time even when the traffic is congested. However, in BPR model, the congestion level (which is equal to the flow/capacity) can go beyond 1, which make the definition of capacity somewhat confusing. Transportation engineers is often using BPR model in conjunction with the capacity derived from Highway Capacity Manual (HCM).

In the scenario file, you can optionally specify the travel time model. If you want to set the travel time model to be Greenshield, then you put the following line in the scenario file:

```
travelTimeModel=Greenshield
```

Without specifying the travel time model, the program will use BPR travel time model as the default model of IFN. If you want to explicitly state that the travel time model is BPR, then you put the following line in the scenario file:

```
travelTimeModel=BPR
```

Capacity Basis

The IFN program is using maximum speed and link capacity in the link file as the basis for the computation. If the maximum speed and capacity fields do not exist in the link file, it would estimate the maximum speed and the link capacity. You can guide this estimation based on one of the following data:

1. Number of lanes per directed link
2. Road width (in meter) per directed link

Either number of lane or road width must be placed in column 5 of the link file.

- If your data in the fifth column of the link file represents the number of lanes, you do not need to specify the capacity basis because the number of lanes is the default value for the capacity basis. Optionally, you can still explicitly state in the scenario file
  ```
  capacityBasis=lanes
  ```
- If your data in the fifth column of the link file represents the road width of each link, then you need to specify in the scenario file
  ```
  capacityBasis=width
  ```

If you need to specify maximum speed and link-capacity (e.g. based on HCM) and not based on the estimation of the IFN program, then you can specify the fields in links file in the following order:

```
LinkNo,Node1,Node2,Dist,Lanes,MaxSpeed,Capacity
```

In this case the Lanes (or Road-width in column 5) of the link data would be ignored and the IFN program will use directly the maximum speed and capacity that you specify.

Cloud Node

For more advance users, you need to know that the IFN requires the network to be strongly connected. If it happens that your network is weakly connected, then you need to create a cloud node and connect each of the source node (or source component) in the network into the cloud node through dummy links and connect the cloud node to each of the sink node (or sink component) in the network using dummy links. When you use a cloud node, you need to specify the node ID of the cloud node. This parameter will affect to hide all the dummy links from showing and the link performance of the dummy links would be set to `nan` (not-a-number). The network performance would be free from the dummy links. If you do not specify the cloud node, the program assume that your network has no cloud node and no dummy links and the network must be strongly connected.

Future Development

There are a lot of fun stuff to develop further and if you have any critics, comments or suggestions to improve, drop me a note. I would welcome your contribution by any means, your programming time, donation or scientific ideas and so on.

Do your part

I hope you find this program useful for your study or work. You can help your own city by setting the base network on your city, and compute the scenarios that most likely will help to solve traffic congestion in your city. Share your ideas in social media and compare it with your friends. Talk with your city government about your ideas.

Bibliography (Scientific Basis)

If you would like to know more about the scientific basis of this work, kindly read and cite any of the following papers:

- Teknomo, K., Gardon, R. and Saloma, C. (2019), Ideal Flow Traffic Analysis: A Case Study on a Campus Road Network, Philippine Journal of Science 148 (1): 51-62.
- Teknomo, K. (2018) Ideal Flow of Markov Chain, Discrete Mathematics, Algorithms and Applications, doi: 10.1142/S1793830918500738
- Teknomo, K. and Gardon, R.W. (2017) Intersection Analysis Using the Ideal Flow Model, Proceeding of the IEEE 20th International Conference on Intelligent Transportation Systems, Oct 16-19, 2017, Yokohama, Japan
- Teknomo, K. (2017) Ideal Relative Flow Distribution on Directed Network, Proceeding of the 12th Eastern Asia Society for Transportation Studies (EASTS), Ho Chi Minh, Vietnam Sept 18-21, 2017.
- Teknomo, K. (2017) Premagic and Ideal Flow Matrices. https://arxiv.org/abs/1706.08856
- Gardon, R.W. and Teknomo, K. (2017) Analysis of the Distribution of Traffic Density Using the Ideal Flow Method and the Principle of Maximum Entropy, Proceedings of the 17th Philippine Computing Science Congress, Cebu City, March 2017
- Teknomo, K. (2015) Ideal Flow Based on Random Walk on Directed Graph, The 9th International collaboration Symposium on Information, Production and Systems (ISIPS 2015) 16-18 Nov 2015, Waseda University, KitaKyushu, Japan.