User guide (English version)

1 Introduction

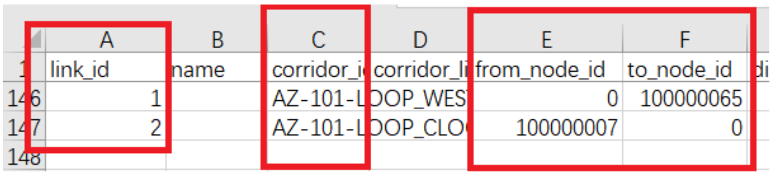
This program is based on the Cell Transmission Model proposed by prof. Daganzo in 1994. This program can read the road network in GMNS format, and at the same time supports the input of time-varying traffic demand, time-varying cell capacity (ie, traffic accidents and other events), and initial cell density.

This program names cells as IP address. A typical cell name is Zoneid.Linkid.Cellid. Zone id is default set as A0, link id is same with which in road network file and cell id starts from C0 to Cx, where x is the number of cell in a link.

2 Input and Output

2.1 Road network file link.csv

Link.csv can be obtained by https://github.com/jiawlu/OSM2GMNS. However, ramp should be defined manually, as figure 1 (a) and (b) shows.



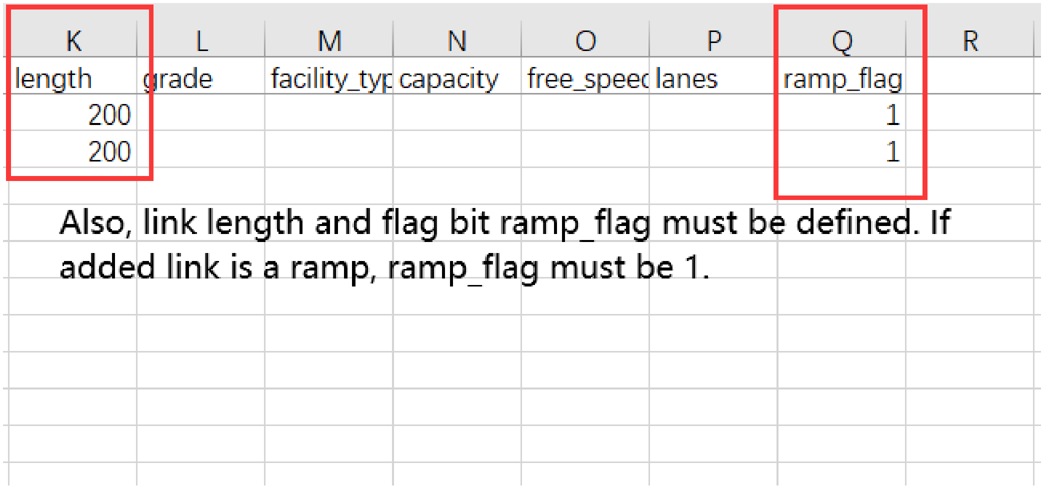
	A	B	C	D	E	F	
1	link_id	name	corridor_i	corridor_l	from_node_id	to_node_id	di
146	1		AZ-101-LOOP_WES		0	100000065	
147	2		AZ-101-LOOP_CLO		100000007	0	
148							

link_id must be defined and must be only.

corridor_id should same with which corridor you want add ramp

If merge cell is going to be added, from_node_id should be defined as link 2 and to_node_id should be 0, vise versa.

(a)



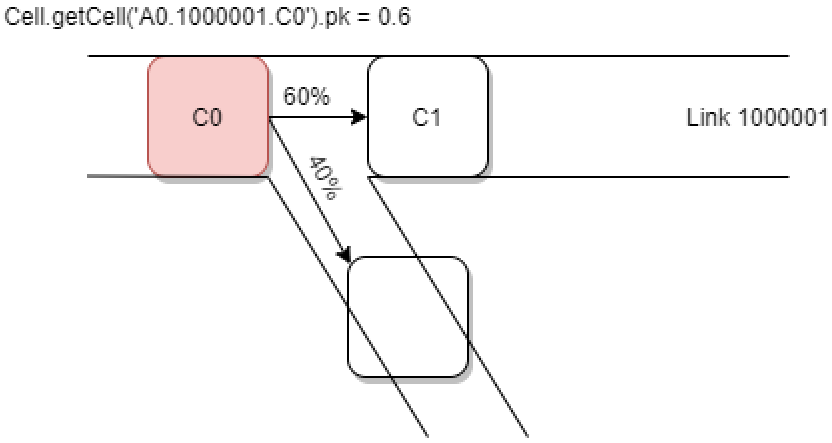
K	L	M	N	O	P	Q	R
length	grade	facility_typ	capacity	free_speec	lanes	ramp_flag	
200						1	
200						1	

Also, link length and flag bit ramp_flag must be defined. If added link is a ramp, ramp_flag must be 1.

(b)

Figure 1 Example of ramp in link.csv

When from_node_id is defined and to_node_id is 0, this link is seen as a merge cell, and when to_node_id is defined and from_node_id is 0, this link is seen as a diverge cell. Default proportion of merge or diverge is main road: ramp = 3:1. If users want to change the proportion, users should change attribute pk of the cell connected with ramp cell. For example, Cell.getCell('A0.1.C1').pk = 0.5.



Cell.getCell('A0.1000001.C0').pk = 0.6

Figure 2 Setting example of proportion of ramp

2.2 Traffic demand file demand.csv

Users can define time varying traffic demand in demand.csv. Demand csv has three columns that time, corridor_id and demand. A variable named time_to_update_demand is defined in function simulation_main(). Users may change value of this variable to change how many time tick elapsed traffic demand changes. If the last row of traffic demand of correspond corridor has read but remaining simulation time is larger than time_to_update_demand, the program will use the last row in demand.csv of correspond corridor as traffic demand in remaining simulation time.

2.3 Supplementary file supply.csv

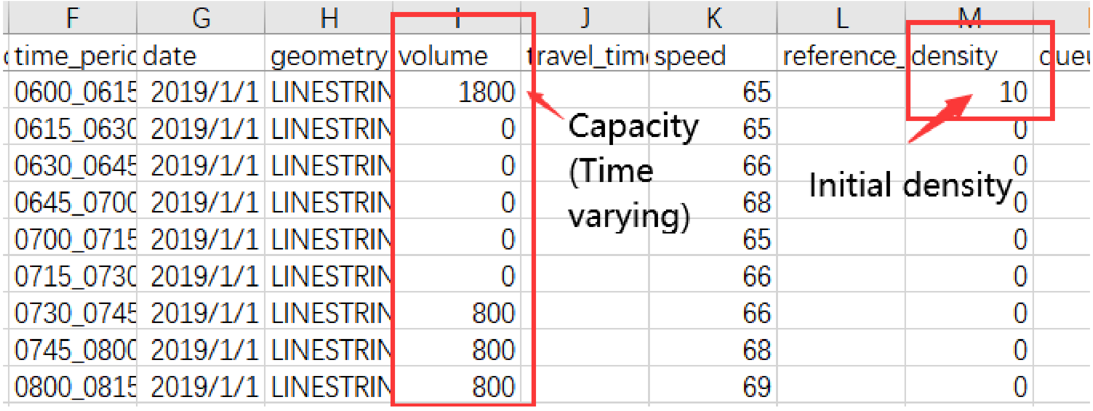
In supply.csv, users should define time_period.

Corridor_id	corridor_link_order	Time_period
11	1	0600_0615
11	1	0615_0630
11	1	0630_0645
11	1	0645_0600

As table above shows, the first row of time_period is 0600_0615, this means that time period is 15 minutes. Total simulation time is 45 minutes.

Users can also define time varying traffic capacity of a link in supply.csv. For example, if volume column in 0615_0630 is defined as 0, this means that this link is completely blocked and capacity of this link is 0.

And initial density can be defined in the first row of a link, as figure 3 shows.



F	G	H	I	J	K	L	M
time_peric	date	geometry	volume	travel_tim	speed	reference_	density
0600_0615	2019/1/1	LINESTRIN	1800		65		10
0615_0630	2019/1/1	LINESTRIN	0		65		0
0630_0645	2019/1/1	LINESTRIN	0		66		0
0645_0700	2019/1/1	LINESTRIN	0		68		0
0700_0715	2019/1/1	LINESTRIN	0		65		0
0715_0730	2019/1/1	LINESTRIN	0		66		0
0730_0745	2019/1/1	LINESTRIN	800		66		0
0745_0800	2019/1/1	LINESTRIN	800		68		0
0800_0815	2019/1/1	LINESTRIN	800		69		0

Figure 3 Setting example of capacity and initial density

2.4 Output files

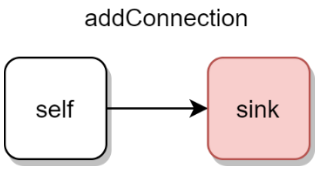
Output files contain density profile and flow profile.

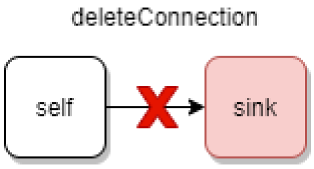
3 Attributes and method of Cell class

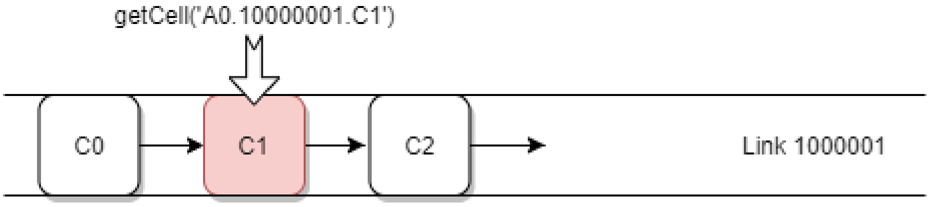
3.1 Attributes

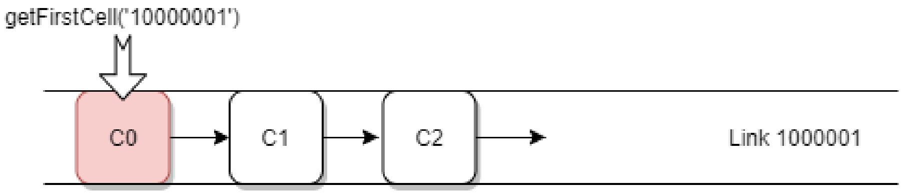
Attribute name	Meaning	Default value
Kjam	Jam density	220 veh/km
Qmax	Capacity	2160 veh/h
Vf	Free flow speed	60 km/h
w	Shock wave speed	12 km/h
cellid	Cell id	N/A
linkid	Link id	N/A
zoneid	Zone id	N/A
updated	Represent if density of the cell has been updated	False
Time_interval	Time tick of simulation	6 seconds
Arr_rate	Arrival rate of a cell	0 veh/h
Dis_rate	Discharge rate of a cell	2160 veh/h
Length	Cell length	M
pk	Proportion of merge or diverge	0.75
Ramp_flag	Flag bit represent whether a link is ramp	0

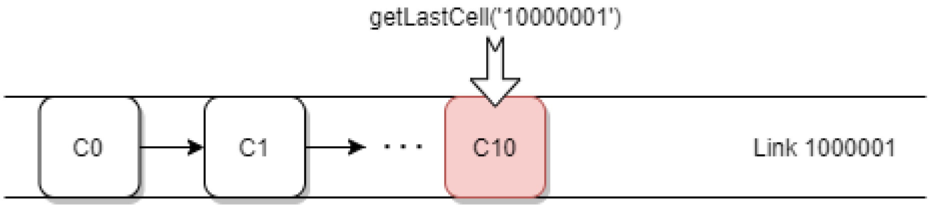
3.2 Method

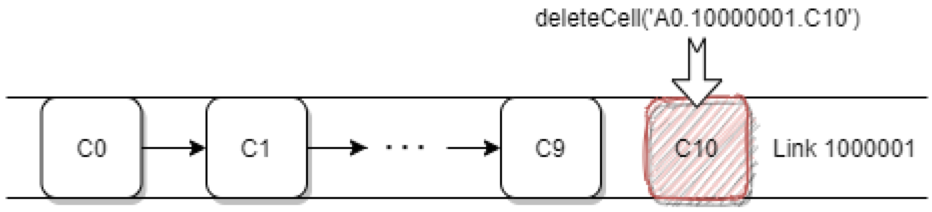
Method	addConnection(self, sink)
Input parameter(s)	Cell Class instance
Return(s)	None
Add a connection for Cell Class instance called the method with another Cell Class instance that from parameter sink.	
	

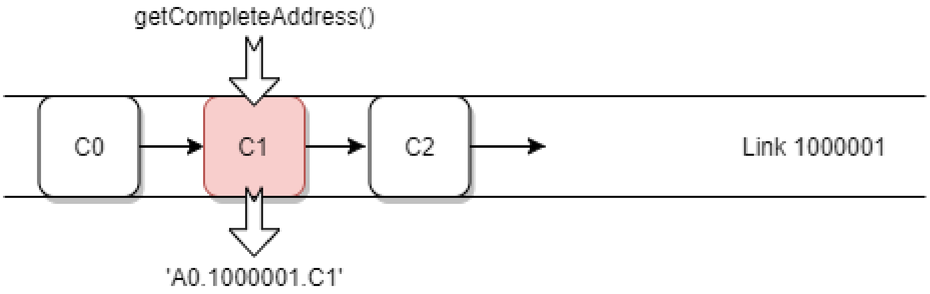
<table>
<tr><td>Method</td><td>deleteConnection(self, sink)</td></tr>
<tr><td>Input parameter(s)</td><td>Cell Class instance</td></tr>
<tr><td>Return(s)</td><td>None</td></tr>
<tr><td colspan="2">Delete a connection for Cell Class instance called the method with another Cell Class instance that from parameter sink.</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

<table>
<tr><td>Method</td><td>getCell(cid)</td></tr>
<tr><td>Input parameter(s)</td><td>String cid, which is complete address of a cell</td></tr>
<tr><td>Return(s)</td><td>Cell Class instance</td></tr>
<tr><td colspan="2">According to complete address of a cell， return correspond Cell Class instance.</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

<table>
<tr><td>Method</td><td>getFirstCell(linkid)</td></tr>
<tr><td>Input parameter(s)</td><td>String linkid, id of a link</td></tr>
<tr><td>Return(s)</td><td>Cell Class instance</td></tr>
<tr><td colspan="2">Return the first Cell Class instance that correspond to input link id.</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

<table>
<tr><td>Method</td><td>getLastCell(linkid)</td></tr>
<tr><td>Input parameter(s)</td><td>String linkid, id of a link</td></tr>
<tr><td>Return(s)</td><td>Cell Class instance</td></tr>
<tr><td colspan="2">Return the last Cell Class instance that correspond to input link id.</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

Method	deleteCell(cid)
Input parameter(s)	String cid, which is complete address of a cell
Return(s)	None
Delete a Cell Class instance.	
 deleteCell('A0.10000001.C10') C0 C1 C9 C10 Link 1000001	

Method	getCompleteAddress(self)
Input parameter(s)	None
Return(s)	String
For a Cell Class instance called the method, return complete address of the cell.	
 getCompleteAddress() C0 C1 C2 Link 1000001 'A0.1000001.C1'	

4 FAQ

Q: Why I can’t get a cell by getCell method?

A: This may be caused by how the program read link.csv. If your link id purely consists with number, such as 100001, then correspond link id in the program is ‘100001.0’. That is a float number is read and is converted to string directly. So that you may get cell by such form: getCell(‘A0.100001.0.C0’)

Q: Why exception raises when use addConnection method?

A: A cell can only connect or be connected with no larger than two other cells, and a cell cannot connect with merge cell and diverge cell simultaneously.